Non-uniform Consistency of Online Learning with Random Sampling

Changlong Wu WUCHANGL@HAWAII.EDU
University of Hawaii at Manoa

Narayana Santhanam NSANTHAN@HAWAII.EDU
University of Hawaii at Manoa

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We study the problem of online learning a hypothesis class and a given binary 0-1 loss function, using instances generated $i.i.d.$ by a given distribution. The goal of the online learner is to make only finitely many errors (loss 1) with probability 1 in the infinite horizon. In the binary label case, we show that hypothesis classes are online learnable in the above sense if and only if the class is *effectively* countable. We extend the results to hypothesis classes where labels can be non-binary. Characterization of non-binary online learnable classes is more involved for general loss functions and is not captured fully by the countability condition even for the ternary label case.

In the computational bounded setup, we compare our results with well known results in *recursive function* learning, showing that the class of all total computable functions is indeed learnable with computable online learners and randomized sampling. Finally, we also show that the finite error guarantee will not be affected even when independent noise is added to the label.

Keywords: Online Learning, Finite Error, Almost Surely, Consistency

1. Introduction

Let $\mathcal{H}$ be a set of functions from $\mathcal{X} \to \{0,1\}$, where $\mathcal{X}$ is an instance space. The online learning setup is a game between two parties, Nature and the learner, both of whom know the class $\mathcal{H}$. Nature chooses a function $h \in \mathcal{H}$ at the beginning of the game. The game proceeds in steps. At each step n, the learner is provided with an instance $x_n \in \mathcal{X}$. The learner has to guess the label $h(x_n)$, potentially using the history built up till that step. Subsequently, the true label $h(x_n)$ is revealed to the learner, and the learner incurs a binary loss—1 if her guess differs from $h(x_n)$ that indicates an error, 0 for no error. The goal of the learner is a strategy that minimizes the number of errors made.

In his seminal work, Littlestone (1988) showed that the number of errors is bounded by what is known as the Littlestone dimension $\mathrm{Ldim}(\mathcal{H})$ of $\mathcal{H}$. This dimension is independent of the number of steps the game goes on for. When the class $\mathcal{H}$ is finite, Littlestone (1988) proved that the number of errors is bounded by $\log|\mathcal{H}|$ using a halving argument.

However, the Littlestone dimension can be too restrictive a measure to handle really rich classes that may not have finite Littlestone dimension, such as the class **P** of all polynomial time decidable problems. One common approach that circumvents this difficulty considers the *non-realizable* setup, where the goal of learner is competitive optimality with respect to a reference class $\mathcal{H}$. Ben-David et al. (2009) showed that one can achieve an expected $O(\sqrt{T\log(T)\mathrm{Ldim}(\mathcal{H})})$ regret in a horizon T game using a weighted-majority algorithm. See (Shalev-Shwartz and Ben-David, 2014, Chapter 21) and the references therein for more results in this line. However, the regret bound doesn't really tell how many *actual* errors the learner could make.

One common approach that deal with rich classes in the realizable case is to relax the requirement of uniform bound to point-wise bounds that could depend on individual hypotheses in the class. Perhaps the most well-known setup of this flavor in learning theory literature is non-uniform PAC learnability and the concept of Structural Risk Minimization by Vapnik and Chervonenkis (1974); Vapnik (2013). A more related scenario in the online learning setup was studied in the theory of learning recursive functions. In this setup, one assumes the class $\mathcal{H}$ to be functions from $\mathbb{N} \to \{0,1\}$, and the instances are presented sequentially as $\{1,2,\cdots\}$. A class $\mathcal{H}$ is said to be recursively learnable if there exist a computable learner such that the number of errors is *finite*, no matter what model is selected from $\mathcal{H}$. See Zeugmann and Zilles (2008) for a survey on this topic.

This paper generalizes the deterministic finite error setting in the recursive function learning literature with a more realistic randomized setup. Instead of the learner observing instances in a predetermined order, we assume that they are sampled from a distribution μ over $\mathcal{X}$ independently at each time step. A class $\mathcal{H}$ of functions from $\mathcal{X} \to \{0,1\}$ is *eventually almost surely* (or eas for abbreviation) online learnable with respect to μ, if there exists an online learner that makes only finite many errors with probability 1, no matter what function is choosen from $\mathcal{H}$.

The conceptual basis of our problem setup lies in the *almost sure hypothesis testing* framework studied in the statistics and information theory communities. See Cover (1973); Dembo and Peres (1994); Kulkarni and Tse (1994); Santhanam and Anantharam (2015); Wu and Santhanam (2019) for a sample of results along this line in hypothesis testing, classification and prediction. There has been work in this line along the computability angle as well, see for example Leshem (2006). We make a note of other related work on an infinite horizon machine learning framework Mitchell et al. (2018), the universal prediction framework in Merhav and Feder (1998), and randomized online learning scenario in Haussler et al. (1994) as well. Novel modifications to the Littlestone dimension have also been considered, see for example, Sayedi et al. (2010); Li et al. (2011); Zhang and Chaudhuri (2016).

Characterization Our first result fully characterizes eas-online learnability in the binary label case. We show that for all instance spaces $\mathcal{X}$ satisfying a regularity condition (satisfied by most common spaces, including *e.g.* the Borel space over $\mathbb{R}^d$) and distributions μ with potentially infinite support, a class $\mathcal{H}$ of *measurable* functions from $\mathcal{X} \to \{0,1\}$ is eas-online learnable if and only if $\mathcal{H}$ is *effectively* countable. Informally, a class $\mathcal{H}$ is countable if we do not distinguish between hypotheses that agree with probability 1 under μ (see Definition 3 for a more complete definition).

Applying our characterization to the class $\mathcal{T}$ of all linear threshold functions from $[0,1] \to \{0,1\}$, i.e. functions of form

$$h_a(x) = \begin{cases} 1, \text{ if } x \geq a \\ 0, \text{ otherwise,} \end{cases}$$

shows that $\mathcal{T}$ above is *not* eas-online learnable for any probability measure over $[0,1]$ that admits a continuous density. However, the class $\mathcal{T}$ has VC dimension 1. For a uniform distribution over $[0,1]$, an empirical risk minimization approach will result in an online learning strategy for the linear threshold functions. The probability of making an error with this approach at time step n is at most $O(\frac{\log n}{n})$. Because $\mathcal{T}$ is not effectively countable, and hence not eas-online learnable, we can conclude immediately that the empirical risk minimization bound cannot be improved from $\frac{\log n}{n}$ to e.g. $\frac{1}{n \log^{1+\epsilon} n}$ with any $\epsilon > 0$. If it could, the Borel-Cantelli lemma would imply that the class $\mathcal{T}$ with uniform distribution would be eas-online learnable, since $1/n \log^{1+\epsilon} n$ is summable over n.

The lower bound only differs by a poly $\log(n)$ factor compared to the optimal $1/n$ lower bound as showed in Schuurmans (1997), and we only needed to observe that the set of all linear threshold functions is not effectively countable to infer this result!

Countable label spaces Departing from the basic result, we show that the characterization of online learning still holds for classification loss even if the label space is countable. However, when more general binary losses are allowed, there exist distributions with infinite support and effectively uncountable classes that are eas-online learnable, even when the label size is 3.

Computable learners Suppose the learner is required to be computable in addition. We show that there exist distributions over $\mathbb{N}$ with infinite support, such that the class of all total computable functions from $\mathbb{N} \to \{0, 1\}$ is eas-online learnable with a computable learner. This is in contrast to the sequential sampling scenario of recursive learning, see Zeugmann and Zilles (2008).

Noisy labels We finally consider online learning with noisy labels, and show that effectively countable classes with binary labels are still online learnable when labels are corrupted with independent Bernoulli noise. Surprisingly, we show that one could also learn effectively countable classes with binary labels when the instances are presented sequentially with noisy labels and in deterministic order. Here we would only see any instance only once with a noisy label, yet we could leverage the noisy labels from different instances to learn the underlying hypothesis.

2. Problem setup

Let $\mathcal{X}$ be the instance space that endowed with some fixed *separable* σ-algebra $\mathcal{F}$. We say $\mathcal{F}$ to be separable, if for all probability distribution μ on $\mathcal{F}$, we have a countable set $\mathcal{G}$ of measurable sets in $\mathcal{F}$, such that for any measurable set $A \in \mathcal{F}$ and $\epsilon > 0$, there exist $G_\epsilon \in \mathcal{G}$ such that

$$\mu(A \Delta G_\epsilon) \leq \epsilon,$$

where Δ is symmetric difference. Clearly, the Borel σ-algebra over $\mathbb{R}^d$ is separable (e.g. we can take $\mathcal{G}$ to be the algebra generated by the set of all cuboids in $\mathbb{R}^d$ with vertices at rational coordinates). We will also assume the single point set of $\mathcal{X}$ is measurable.

Let $\mathcal{Y}$ be a label space that is often assumed to be finite or countable. A binary loss is a function $\ell : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ that is symmetric and reflexive, i.e. $\ell(y_1, y_2) = \ell(y_2, y_1)$ and $\ell(y_1, y_1) = 0$ for all $y_1, y_2 \in \mathcal{Y}$. And a learning strategy is function

$$\Phi : (\mathcal{X} \times \mathcal{Y})^* \times \mathcal{X} \to \mathcal{Y},$$

that is measurable over cylinder σ-algebra on $(\mathcal{X} \times \mathcal{Y})^\infty \times \mathcal{X}$.

We consider the following learning game between Nature and a learner that proceeds by time steps. Let $\mathcal{H}$ be a class of measurable functions from $\mathcal{X} \to \mathcal{Y}$ and μ be a probability measure over $\mathcal{X}$, which are known to all parties. At the beginning, the nature chooses a function $h \in \mathcal{H}$. At each time step n, the nature independently samples $X_n \sim \mu$ and provides it to the learner. The learner then outputs an estimate Y_n of the true label $h(X_n)$, potentially using the history $\{(X_1, h(X_1)), \cdots, (X_{n-1}, h(X_{n-1}))\}$ thus far. Nature then provides the true label $h(X_n)$ to the learner, and the learner incurs the binary loss $\ell(Y_n, h(X_n)$. The learner makes an *error* at time step n if $\ell(y_n, h(x_n)) = 1$.

Denote $Z_i = (X_i, h(X_i))$ to be the instance-label pair at time step i, and $Z_1^i = (Z_1, \cdots, Z_i)$ is the history observed by the learner upto time step i.

Definition 1 *A class $\mathcal{H}$ is said to be eas-online learnable w.r.t. distribution μ, if there exists a learning strategy Φ such that*

$$\Pr\left(\sum_{n=1}^{\infty}\ell(\Phi(Z_1^{n-1},X_n),h(X_n))<\infty\right)=1,$$

for all $h\in\mathcal{H}$ with $X_1,X_2,\cdots\overset{i.i.d.}{\sim}\mu$.

We also introduce the following stronger version of eas-learnability when the distribution is unknown to the learner.

Definition 2 *A class $\mathcal{H}$ is said to be strongly eas-online learnable, if there exists a learning strategy Φ such that*

$$\Pr\left(\sum_{n=1}^{\infty}\ell(\Phi(Z_1^{n-1},X_n),h(X_n))<\infty\right)=1,$$

for all $h\in\mathcal{H}$ and μ with $X_1,X_2,\cdots\overset{i.i.d.}{\sim}\mu$.

In both cases above, we will say that the strategy Φ eas-online learns (or strongly eas-online learns) $\mathcal{H}$ w.r.t. μ. For notational convenience, we drop the reference to μ where doing so leads to no ambiguity.

The following notion will be used frequently in this paper.

Definition 3 *A class $\mathcal{H}'$* effectively covers *a class $\mathcal{H}$ w.r.t. distribution μ and loss ℓ, if for all $h\in\mathcal{H}$ there exists $h'\in\mathcal{H}'$ such that*

$$\mu\{x:\ell(h(x),h'(x))=1\}=0.$$

We say $\mathcal{H}$ is effectively countable *(respectively size n) w.r.t. μ and ℓ, if there exist $\mathcal{H}'$ that is countable (respectively size n), such that $\mathcal{H}'$ effectively covers $\mathcal{H}$ w.r.t. μ and ℓ.*

We begin with the following lemma, an online learning version of Structural Risk Minimization.

Lemma 4 *Let $\mathcal{H}_1,\mathcal{H}_2,\cdots$ be countably function classes that share the same instance space and loss function. If $\mathcal{H}_n$ is eas-online learnable for all $n\in\mathbb{N}$ w.r.t. distribution μ, then*

$$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$$

is also eas-online learnable w.r.t. μ.

Proof Let Φ_k to be the strategy that eas-online learns class $\mathcal{H}_k$. We define a strategy for $\mathcal{H}$ as follows. At each time step n, denote by $e(n,k)$, the number of errors that Φ_k would made if it were employed in the first $n-1$ time steps. Let

$$J_n=\operatorname*{argmin}_{k\in\mathbb{N}}\{e(n,k)+k\}. \tag{1}$$

We then use Φ_{J_n} to make the prediction at time step n.

We now show that this strategy indeed makes finitely many errors with probability 1 no matter what h is chosen from $\mathcal{H}$. Assume $h \in \mathcal{H}_t$ for some $t \in \mathbb{N}$. Let $B \subset \mathcal{X}^\infty$ be the set such that Φ_t would make finite errors on all realizations in B. We have $\mu(B) = 1$ because Φ_t eas-online learns $\mathcal{H}_t$. Fix any $\mathbf{x} \in B$ and denote s to be the number of errors Φ_t would make on $\mathbf{x}$. We have

$$e(n,t) + t \leq s + t$$

for all n and from (1), note that therefore for all n, J_n, the index chosen is $\leq s + t$.

Furthermore, for any Φ_i with $i \in \{1, \cdots, s+t\}$, once Φ_i makes more than $s + t$ errors, it will no longer be chosen in (1). Therefore, we will make at most $(s+t)^2$ errors, and thus finitely many errors on $\mathbf{x}$. Therefore, our strategy also makes finite errors with probability 1 when h is selected. The lemma follows. ■

Remark 5 *Note that Lemma 4 holds for strong eas-online learning as well, since the construction of learning strategy does not depend on the knowledge of underlying distribution.*

We prove the following technical lemma that relates deterministic sampling with randomized sampling.

Lemma 6 *Let $a_1, a_2, \cdots$ be an arbitrary ordering of $\mathbb{N}$, and $s_1, s_2, \cdots \in \mathbb{N}^+$ be an arbitrary sequence. Then there exist a distribution p over $\mathbb{N}$, such that*

$$p\left(\exists N : \forall n \geq N,\ T_n - T_{n-1} \geq s_{n-1}\right) = 1$$

where T_n is the first time a_n appears in an $i.i.d.$ sample from p.

Proof Let $p_n = p(X = a_n)$. Since $T_n - T_{n-1} < s_{n-1}$ implies that a_n appears earlier than the s_{n-1}'th appearance of a_{n-1}, we have

$$p(T_n - T_{n-1} < s_{n-1}) \leq 1 - \left(\frac{p_{n-1}}{p_n + p_{n-1}}\right)^{s_{n-1}}. \tag{2}$$

Taking

$$p_n = \frac{1}{(n!)^2 \prod_{i=1}^{n-1} s_i},$$

we have that (2) is further upper bounded by $\frac{1}{n^2}$. Since $\frac{1}{n^2}$ is summable, an application of the Borel-Cantelli Lemma completes the proof. ■

3. Binary label

In this section, we will consider the case when the label is binary and the loss is $\ell(y_1, y_2) = 1\{y_1 \neq y_2\}$. We will refer to this loss as the *classification loss* in the sequel. Note that since our loss function is binary valued on binary labels, the only losses can either be trivial or classification loss. We say a distribution over $\mathcal{X}$ to be *non-degenerate* if it has infinite support.

The following theorem fully characterizes the eas-online learnability in the binary label case.

Theorem 7 *A class $\mathcal{H}$ with binary labels is eas-online learnable w.r.t. distribution μ iff $\mathcal{H}$ is* effectively *countable w.r.t. μ.*

Proof To see that if $\mathcal{H}$ is effectively countable, it is eas-online learnable, note that for any class that contains *effectively* 1 hypothesis is trivially eas-online learnable. Therefore, any class $\mathcal{H}$ with effectively countably many hypotheses is eas-online learnable by appealing to Lemma 4.

For the necessary part of the proof, we will assume w.l.o.g. that for all $h_1 \neq h_2 \in \mathcal{H}$ we have $\Pr_{X\sim\mu}(h_1(X) \neq h_2(X)) > 0$. Note that any hypothesis class can be reduced to one satisfying the above property by choosing a representative function in each equivalence class defined by the relation $h_1 \sim h_2 : \Pr_{x\sim\mu}(h_1(x) \neq h_2(x)) = 0$.

We now prove that if $\mathcal{H}$ admits a function $\Phi : (\mathcal{X} \times \{0,1\})^* \times \{0,1\} \to \{0,1\}$ such that $\forall h \in \mathcal{H}$

$$\Pr\left(\sum_{n=1}^{\infty} 1\{\Phi(Z_1^{n-1}, X_n) \neq h(X_n)\} < \infty\right) = 1, \tag{3}$$

where $Z_1^{n-1} = (Z_1, \cdots, Z_{n-1})$ and $Z_i = (X_i, h(X_i))$ is the instance-label pair at step i generated by μ, then $\mathcal{H}$ is countable.

Define the event

$$A_n^h = \left\{ X_1^\infty : \sum_{k=n}^{\infty} 1\{\Phi(Z_1^k, X_{k+1}) \neq h(X_{k+1})\} > 0 \right\},$$

and let

$$\mathcal{H}_n = \left\{ h \in \mathcal{H} : \Pr\left(A_n^h\right) \leq \frac{1}{4} \right\}. \tag{4}$$

From equation (3), we must have $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$. Since for any $h \in \mathcal{H}$, we have $\Pr(A_n^h) \to 1$ as $n \to \infty$, there exist some k such that $\Pr(A_k^h) \leq \frac{1}{4}$, i.e. $h \in \mathcal{H}_k$. We show that $\mathcal{H}_n$ is countable for all $n \in \mathbb{N}$, which will prove the Theorem.

Central to our proof is our claim that for all $n \in \mathbb{N}$ we have

$$\inf\{\Pr_{X\sim\mu}(h_1(X) \neq h_2(X)) : h_1 \neq h_2 \in \mathcal{H}_n\} > 0. \tag{5}$$

Now equation (5) implies that $\mathcal{H}_n$ is countable. To see this, let

$$d(h_1, h_2) = \Pr_{X\sim\mu}(h_1(X) \neq h_2(X)).$$

The σ-algebra over $\mathcal{X}$ is separable, so by definition, there is a countable dense subset of the $\sigma-$algebra with respect to the distance d. Since each set in the $\sigma-$algebra can be represented by a measurable function from $\mathcal{X} \to \{0,1\}$, we conclude that there is a countable collection $\mathcal{G} = \{g_1, g_2 \cdots\}$ of measurable functions from $\mathcal{X} \to \{0,1\}$, such that $\mathcal{G}$ is dense in the set of all measurable functions from $\mathcal{X} \to \{0,1\}$ under the distance d defined above.

Suppose the infimum in (5) is $\epsilon > 0$. Since $\mathcal{G}$ is dense in the the set of all binary valued measurable functions on $\mathcal{X}$, for each $h \in \mathcal{H}_n$, there is a function $g \in \mathcal{G}$ such that $d(h, g) < \epsilon/2$. By the triangle inequality, for all $h' \neq h$, we will also have $d(h', g) > \epsilon/2$. Therefore every $h \in \mathcal{H}_n$ can be associated with a distinct $g \in \mathcal{G}$, which then implies that $\mathcal{H}_n$ is at most countable.

We now establish equation (5). The intuition behind this claim is that if the infimum in (5) were 0, then we could choose two hypotheses in $\mathcal{H}_n$ that were arbitrarily close in the distance d—and

hence close enough that they are indistinguishable with sample size n on a large enough subset A. However, these hypothesis will eventually differ in the suffix of the strings in A, therefore no predictor could agree with both of them in the suffix past step n. But if the probability of A is large enough, it contradicts the definition of $\mathcal{H}_n$ in (4).

Formally, suppose (5) did not hold. Then there exist $h_1, h_2 \in \mathcal{H}_n$ such that $0 < d(h_1, h_2) < \delta_n$, where δ_n is chosen so that $(1-\delta_n)^n > \frac{1}{2}$.

Let $A \subset \mathcal{X}^\infty$ be the event that h_1, h_2 cannot be distinguished with n samples. By construction, we have $\Pr(A) > \frac{1}{2}$. For $j \in \{1, 2\}$, let

$$p_j = \Pr(\Phi \text{ makes error after step } n \text{ on } h_j).$$

We show that $\max\{p_1, p_2\} > \frac{1}{4}$, which will then contradict equation (4) since $h_1, h_2 \in \mathcal{H}_n$, thus establishing equation (5). To see that $\max\{p_1, p_2\} > \frac{1}{4}$, we use a probabilistic argument. Let $\mathbf{h}$ be the random variable uniformly chosen from $\{h_1, h_2\}$. We only need to show

$$\mathbb{E}_{\mathbf{h}}\mathbb{E}_{X\sim\mu^\infty}[1\{\Phi \text{ makes error after step } n \text{ on } \mathbf{h}\} \mid X \in A] \geq \frac{1}{2} \tag{6}$$

Let $B \subset \mathcal{X}^\infty$ be the event that there exists a instance Y after step n that first reveals $h_1(Y) \neq h_2(Y)$. We have $\Pr(B) = 1$, since $d(h_1, h_2) > 0$. Note that

$$\mathbb{E}_{X\sim\mu^\infty}[\mathbb{E}_{\mathbf{h}}[1\{\Phi \text{ makes error after step } n \text{ on } \mathbf{h}\}] \mid X \in A \cap B] \geq \frac{1}{2}, \tag{7}$$

since condition on any $X \in A \cap B$ event $C = \{\Phi \text{ makes error at sample } Y \text{ on } \mathbf{h}\}$ implies the event in the equation above, and because $\mathbb{E}_{\mathbf{h}}[1\{C\}] = \frac{1}{2}$. We now have

$$\mathbb{E}_{X\sim\mu^\infty}[\mathbb{E}_{\mathbf{h}}[1\{\Phi \text{ makes error after step } i \text{ on } \mathbf{h}\}] \mid X \in A] \geq \frac{1}{2}, \tag{8}$$

since $\Pr(B) = 1$. Finally (8) implies (6) by exchanging order of expectation, which is justified by Fubini's theorem. ■

Example 1 *Let $\mathcal{X} = [0, 1]$ and μ be the uniform distribution over $[0, 1]$. We consider the linear threshold functions*

$$h_a(x) = 0 \text{ if } a \geq x \text{ and } h_a(x) = 1 \text{ otherwise}.$$

Denote $\mathcal{H} = \{h_a : a \in [0, 1]\}$. By Theorem 7 we know that $\mathcal{H}$ is not *eas-online learnable.*

A couple of observations need to be emphasized here. Note that the VC dimension of $\mathcal{H}$ is 1. Therefore the VC-theorem (Shalev-Shwartz and Ben-David, 2014, Thm 6.7) posits that there is a learning rule $\hat{h}_n$, such that for any $h_a \in \mathcal{H}$ and a sample S_n of size n, with high probability say $1 - \frac{1}{n^2}$ over S_n, we have

$$\Pr_{x\sim\mu}[\hat{h}_n(S_n, x) \neq h_a(x)] \leq O\left(\frac{\log n}{n}\right).$$

Theorem 7 implies that we cannot improve the upper bound from $O\left(\frac{\log n}{n}\right)$ to, say, $O\left(\frac{1}{n \log^{1+\epsilon} n}\right)$ with $\epsilon > 0$. If we could, note that since $\sum_{n=1}^{\infty} \frac{1}{n \log^{1+\epsilon} n} < \infty$, an application of the Borel-Cantelli lemma would imply that $\hat{h}_n$ only makes a finite number of errors no matter the hypothesis in force, thus violating Theorem 7.

We observe the following simple corollary.

Corollary 8 *If a binary measurable function class $\mathcal{H}$ has finite Littlestone dimension, then $\mathcal{H}$ is effectively countable w.r.t. any distribution. In particular, if the domain is countable, then $\mathcal{H}$ is countable.*

Proof Finite Littlestone dimension implies $\mathcal{H}$ is online learnable with bounded number of errors and an adversarial sampling process (Shalev-Shwartz and Ben-David, 2014, Chapter 21). The corollary follows by Theorem 7 and notice that the learning strategy is independent of the distribution. The second part follows by the fact that for any hypothesis class $\mathcal{H}$ over a countable domain $\mathcal{X}$, if $\mathcal{H}$ is effectively countable w.r.t. a distribution with supported of the whole of $\mathcal{X}$, then $\mathcal{H}$ is countable. ■

The proof of the necessary condition of Theorem 7 also yields that if a class $\mathcal{H}$ is strongly eas-online learnable, then $\mathcal{H}$ is effectively countable w.r.t. any distribution. By Corollary 8 above, this implies that the restriction of $\mathcal{H}$ on any countable subset of $\mathcal{X}$ must be countable. However, this doesn't mean that the class $\mathcal{H}$ have to be countable. To see this, consider the class $\mathcal{T}$ of all functions $t_a : [0,1] \to \{0,1\}$ with $a \in [0,1]$ that has the following form:

$$t_a(x) = \begin{cases} 1, & \text{if } x = a \\ 0, & \text{otherwise} \end{cases} .$$

It is easy to see that $\mathcal{T}$ is strongly eas-online learnable (in fact has Littlestone dimension 1), but the class is uncountable. We leave it as an open problem to determine whether a class that is eas-online learnable w.r.t. any distribution is also strongly eas-online learnbale. (Note that, in the former setting the learning strategy can be different for different distribution, while the strongly eas-online learnable requires a universal learning rule.)

4. Multi-Class label

We consider the case when the output set $\mathcal{Y}$ has more than 2 elements and we assume the loss ℓ to be an arbitrary function. Before analyzing the randomized setting, we consider a simpler deterministic setup. A deterministic sampling process in the most general setting is a function $\mathcal{S} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}$, which select the next sample based on the previous sample-label pairs (but not the learner's answer). For any class $\mathcal{H}$ and process $\mathcal{S}$, one can construct an infinite $|\mathcal{Y}|$-nary tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with a label function L (for both vertices and edges) such that

1. Each edge has a label in $\mathcal{Y}$ and the label of edges with same parent are distinct.
2. Each vertex in $\mathcal{T}$ has a label in $\mathcal{X}$. The root v_0 has label $L(v_0) = \mathcal{S}(\epsilon)$, where ϵ is empty string. For each vertex $v \in \mathcal{T}$, we denote $v_0 \to v_1 \to \cdots v_n = v$ to be the unique path from root to v. Let $z_i = (L(v_{i-1}), L(v_{i-1} \to v_i))$, the label of v is given by $L(v) = \mathcal{S}(z_1, \cdots, z_n)$.
3. Each vertex v in $\mathcal{T}$ has a child u such that $L(v \to u) = y$ if and only if $\exists h \in \mathcal{H}$ such that $h(L(v_i)) = L(v_i \to v_{i+1})$ for all $0 \le i \le n$ and $h(L(v)) = y$, where v_i is defined in item 2.

Clearly, every function $h \in \mathcal{X}$ will associated with a unique infinite path $v_0 \to v_1^h \to v_2^h \to \cdots$ in $\mathcal{T}$ such that $h(L(v_i^h)) = L(v_i^h \to v_{i+1}^h)$. However, not every infinite path will have a function in $\mathcal{H}$ associated with it. The learning process can be viewed as traversing an infinite path in $\mathcal{T}$. Such a tree is also known as the mistake tree in the binary online learning literature, see Zhang and Chaudhuri (2016). A valuation of $\mathcal{T}$ is a function $\mathbf{v} : \mathcal{V} \to \mathbb{N}$ such that for each $v \in \mathcal{V}$ we have

1. $\mathbf{v}(v) \geq \max_{u \in C(v)}\{\mathbf{v}(u)\}$, where $C(v)$ is the children vertex set of v;
2. $\mathbf{v}(v) \geq 1 + \max_{u \in C(v)}\{\mathbf{v}(u)\}$ if $L_v^M = \{L(v \to u) : u \in C^M(v)\}$ cannot be covered by $\mathcal{Y}$, where $C^M(v)$ is the set of children vertex of v that has maximal value. We say a set $A \subset \mathcal{Y}$ can be covered by $\mathcal{Y}$ if there exists $y \in \mathcal{Y}$ such that $\forall x \in A,\ \ell(y, x) = 0$.

Note that the valuation may not always exist. The following theorem relates the existence of valuation to the bounded error guarantee.

Theorem 9 *A class $\mathcal{H}$ is online learnable with $\leq B$ errors and a deterministic sampling process $\mathcal{S}$ if and only if there exists a valuation $\mathbf{v}$ on $\mathcal{T}$ such that $\mathbf{v}(v_0) \leq B$, where v_0 is the root of $\mathcal{T}$.*

Proof If $\mathcal{T}$ has a valuation such that $\mathbf{v}(v_0) \leq B$, we simply predict the element that would cover L_v^M at vertex v in the traversing (if no such element, predict anything). Note that the learner makes an error only if the value of the child vertex is reduced by at least 1. Therefore, there will be at most B errors.

If $\mathcal{H}$ is online learnable w.r.t. $\mathcal{S}$ with at most B errors, we define the value at vertex v to be the maximum number of errors that the learning rule could make on the subtree rooted by v.

To prove this is a valid valuation, note that condition 1 can be verified easily. To see condition 2 holds as well, we resort to a proof by contradiction. If condition 2 does not hold at some vertex v, i.e. we have $\mathbf{v}(v) = \max_{u \in C(v)}\{C(v)\}$ but L_v^M cannot be covered by $\mathcal{Y}$. We can choose a path to a child in $C^M(v)$ that has label differs from the prediction given by the learner, thus incuring $\max_{u \in C(v)} \mathbf{v}(u) + 1$ errors starting from v, contradicting our premise. The theorem follows. ■

To capture the *finite* error guarantee, we will need a notion of ranking on the hypotheses in $\mathcal{H}$. A ranking of $\mathcal{H}$ is a function $\mathbf{r} : \mathcal{H} \to \mathbb{N}$. For any vertex $v \in \mathcal{V}$, we denote $\mathcal{H}_v$ to be the hypotheses in $\mathcal{H}$ that share the path from v_0 to v in $\mathcal{T}$, where v_0 is the root of $\mathcal{T}$. For a given ranking $\mathbf{r}$, we denote $\mathcal{H}_v^m$ to be the hypotheses in $\mathcal{H}_v$ that have *minimal* rank, i.e.

$$\mathcal{H}_v^m = \{h \in \mathcal{H}_v : \mathbf{r}(h) = \min\{\mathbf{r}(\mathcal{H}_v)\}\},$$

where $\mathbf{r}(\mathcal{H}_v) = \{\mathbf{r}(h) : h \in \mathcal{H}_v\}$. Now let $L_v^m = \{h(L(v)) : h \in \mathcal{H}_v^m\}$.

We say a class $\mathcal{H}$ to be *rankable* w.r.t. a deterministic sampling process $\mathcal{S}$, if there exists a ranking $\mathbf{r}$ such that for each $h \in \mathcal{H}$ there exists a number N_h, such that for every vertex $v \in \mathcal{V}$ on the infinite path of h with depth larger than N_h, we have $h \in \mathcal{H}_v^m$ and L_v^m is coverable.

Intuitively, the smaller the rank of a function in $\mathcal{H}$, the higher priority we will assign in the learning process. Formally, we prove the following theorem:

Theorem 10 *A class $\mathcal{H}$ is online learnable with finitely many errors w.r.t. a deterministic sampling process $\mathcal{S}$ if and only if $\mathcal{H}$ is rankable w.r.t. $\mathcal{S}$.*

Proof If $\mathcal{H}$ is rankable, we show that $\mathcal{H}$ is online learnable with finitely many errors. The prediction rule works as follows, we predict an element that covers L_v^m at each vertex v in the traversing (if it doesn't exist we predict anything). By definition of rankability, the learner will not make errors after step N_h if the underlying hypothesis is h.

To see the converse, suppose Φ is a online learning rule which makes only finitely many errors no matter what the hypothesis $h \in \mathcal{H}$ is. The rank of each hypothesis h is the number of errors Φ would make against h.

To conclude the proof, we will show that this is a valid ranking. Fix any h and let N_h be the time step of the last error Φ make on h. Suppose there exists some vertex v at depth more than N_h such that L_v^m is not coverable. Now the rule Φ will make more errors on at least one of hypotheses $h' \in \mathcal{H}_v^m$ than it made on h. This is because h and h' share the same prefix up to vertex v and hence have the same error pattern till then, but differ on their label in L_v^m. Thus this hypothesis h' must have a higher rank by definition, and therefore could not have been in $\mathcal{H}_v^m$. ■

Note that both Theorem 9 and Theorem 10 work only for deterministic sampling process where there is one mistake tree. However, in the randomized setup the mistake tree will be different for different realization of the sampling. A sufficient condition is to have a universal ranking that works for almost all realizable mistake trees. The following theorem shows that it will happen when the class is effectively countable. We leave the proof to Appendix A.

Theorem 11 *Let $\mathcal{H}$ be a set of measurable functions from $\mathcal{X} \to \mathcal{Y}$. Then $\mathcal{H}$ is eas-online learnable w.r.t. distribution μ and loss ℓ, if $\mathcal{H}$ is effectively countable w.r.t. μ and ℓ.*

Moreover, the effective countability of $\mathcal{H}$ is necessary for eas-online learnability (w.r.t. any distribution μ) if $\mathcal{Y}$ is countable and ℓ is the classification loss, i.e. $\ell(y_1, y_2) = 1\{y_1 \neq y_2\}$.

Corollary 12 *Let $\mathcal{H}$ be the class of all continuous functions from $[0,1] \to [0,1]$, $\ell = 1\{|y_1 - y_2| > B\}$ for some $B > 0$. Then $\mathcal{H}$ is eas-online learnable w.r.t. the loss ℓ and any distributions.*

Proof By the Stone-Weierstrass theorem, $\mathcal{H}$ can be covered by polynomials with rational coefficients under the supremum norm. Since polynomials with rational coefficients are countable, $\mathcal{H}$ is effectively coverable w.r.t. any distribution. Theorem 11 then implies that $\mathcal{H}$ is eas-online learnable w.r.t. any distribution. ■

Moreover, the covering set in Corollary 12 is independent of the distribution. Therefore, we know that the class of all continuous functions in Corollary 12 is actually strongly eas-online learnable.

We provide the following example, which shows that if we have $|\mathcal{Y}| \geq 3$ and $\mathcal{X} = \mathbb{N}$, then there exists a class $\mathcal{H}$, distribution p and loss ℓ, such that $\mathcal{H}$ is eas-online learnable w.r.t. p and ℓ, but $\mathcal{H}$ is *not* effectively countable. Thus, the condition in Theorem 11 can't be necessary for arbitrary losses and distributions.

Example 2 *Let $\mathcal{Y} = \{0,1,2\}$, we define a loss ℓ as follows: ℓ is symmetric in its two arguments and*

$$\ell(0,1) = 0 \text{ and } \ell(0,2) = \ell(1,2) = 1.$$

We now construct the class $\mathcal{H}$ as follows. Since the domain is $\mathbb{N}$, we will denote each function in $\mathcal{H}$ as infinite sequences in $\{0,1,2\}^\infty$. Let $\mathcal{B} = \{0,1\}^\infty$ be the class of all binary sequences. We define the following transformation T that maps $\{0,1\}^\infty \to \{0,1,2\}^\infty$. For any $\boldsymbol{b} \in \mathcal{B}$, $T(\boldsymbol{b})$ is the sequence that inserts number 2 that follows each appearance of 0 in $\boldsymbol{b}$. For example, if $\boldsymbol{b} = 00100101011\ldots$ then $T(\boldsymbol{b})$ would read $02021020210210211\ldots$. Now, we define

$$\mathcal{H} = \{T(\boldsymbol{b}) : \boldsymbol{b} \in \mathcal{B}\}.$$

We now define the distribution. By Lemma 6, we have distribution p over $\mathbb{N}$ such that the i.i.d. *samples from p will appear in increasing order of $\mathbb{N}$ eventually almost surely—namely for all m*

large enough, the instance $m-1$ will appear prior to instance m with probability 1. Therefore, one may assume, w.l.o.g., that the sample is generated sequentially as $1, 2, 3, \cdots$.

To see that the class $\mathcal{H}$ is eas-online learnable w.r.t. to p and loss ℓ, for any instance $m \in \mathbb{N}$ we simply predict 2 for the label of instance m if the label of instance $m-1$ is 0 and predict 0 if the label of $m-1$ is 1 or 2. It is easy to check that the rule makes only finitely many errors with probability 1.

Clearly, the class $\mathcal{H}$ is uncountable *and for any two different elements $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathcal{B}$ we can find $l \in \mathbb{N}$ such that $\ell(T(\boldsymbol{b}_1)(l), T(\boldsymbol{b}_2)(l)) = 1$. Therefore, we have $\mathcal{H}$ is* not *effectively coverable by any countable set $\mathcal{H}'$.*

Even though the class in Example 2 is not effectively countable, it has a trivial universal ranking that ranks all functions in $\mathcal{H}$ to 0. We conclude this section with the following conjecture.

Conjecture 13 *A class $\mathcal{H}$ is eas-online learnable w.r.t μ iff there exist a universal ranking $\boldsymbol{r}$ of $\mathcal{H}$ such that $\mathcal{H}$ is* rankable *with ranking $\boldsymbol{r}$ for almost all mistake tree generated by μ.*

5. Computable learner

As we have mentioned in the introduction, our setup has a close connection with *learning of recursive functions*, where one considers the domain to be $\mathbb{N}$ and the instances are presented sequentially, but in addition, the learning rule is required to be computable as well.

Definition 14 *A function $h : \mathbb{N} \to \{0, 1\}$ is* computable *if there exists a Turing machine $\boldsymbol{TM}$ such that $\forall n \in \mathbb{N}$, $\boldsymbol{TM}(n)$ halts and outputs $h(n)$, where the number n is in its binary representation.*

Clearly the class of all computable functions from $\mathbb{N} \to \{0, 1\}$ is countable, since there are only countable Turing machines (Arora and Barak, 2009, Chapter 1). We have the following simple corollary that follows from Lemma 4.

Corollary 15 *Let $\mathcal{H}$ be the set of all computable functions from $\mathbb{N} \to \{0, 1\}$, and let μ be an arbitrary distribution over $\mathbb{N}$. Then $\mathcal{H}$ is eas-online learnable using* i.i.d. *samples generated from μ.*

Unfortunately, the learning rule that derived from Lemma 4 cannot be computable for arbitrary distribution as shown in the following theorem of Barzdiņš and Freivald (1972) (restated here in our notation).

Theorem 16 ((Zeugmann and Zilles, 2008, Thm 5)) *Let $\mathcal{H}$ be a class of computable function from $\mathbb{N} \to \{0, 1\}$. Then there exists a computable learner that eas-online learns $\mathcal{H}$ with sequential sampling if and only if there exists a computable function $g : \mathbb{N} \to \mathbb{N}$ such that the time complexity of each function in $\mathcal{H}$ is eventually dominated by $g(n)$.*

It is easy to show that if $\mathcal{H}$ is the class of *all* computable functions then g in Theorem 16 cannot exist by a simple diagonalization argument. Therefore, by Lemma 6 there exists a distribution p such that $\mathcal{H}$ is not computationally eas-online learnable even with *i.i.d.* sampling from p. However, we will show in the following theorem that there are also *non-degenerate* distributions over $\mathbb{N}$ such that the class of all computable functions is indeed computationally eas-online learnable with *i.i.d.* sampling.

Theorem 17 *Let $\mathcal{H}$ be the class of all computable functions from $\mathbb{N} \to \{0,1\}$. Then there exists a non-degenerate distribution (i.e. has infinite support) q such that $\mathcal{H}$ is computationally eas-online learnable w.r.t. q.*

Proof Let $\mathbf{TM}_1, \mathbf{TM}_2, \cdots$ be an fixed enumeration of all Turing machines. The main idea is to construct a distribution q such that for almost all numbers n, n appears much later than any of $\{1, ..., n-1\}$ with probability 1. We choose the gap period between the first appearances of $n-1$ and n to be $\max\{C(h_i(n) : i \leq n\}$, where $C(h_i(n))$ is the computational time for the ith computable function with input n to stop when computing with a feasible Turing machine of smallest index. Such a distribution q exists by Lemma 6.

We now construct the computable predictor using a *back and forth* trick. The predictor goes as follows:

1. Initialize index $I, J = 1$;
2. In the idle period between the appearance of $n-1$ and n, the predictor do the following. It *simulates* the computation of $\mathbf{TM}_I$ on n with one computational step per time step. (For other samples that encountered in that period, one simply predicts the memorized labels.)
3. At time step of first observing n, output the result of the simulation. (Output arbitrary if the simulation has not stopped.) If the the result matches with the true label, keep I, J. Else:
    a. If $I < J$, set $I = I + 1$ and $J = J$;
    b. Else, set $I = 1$ and $J = J + 1$.

Since the underlying function is computable, there exists a Turing machine $\mathbf{TM}_t$ that computes it. Now, by construction the index I changes if and only if the predictor makes an error.

We show that the predictor only makes finitely many errors by a proof by contradiction. If indeed the predictor makes infinite errors, we known that I would repeatedly hit t until the sample is coming sequentially. By construction of the idle time, $\mathbf{TM}_t$ would then finish the computation and make the right prediction in the following times steps, which is a contradiction. ■

Remark 18 *Note that the reason why Theorem 17 does not contradict to the impossibility result implied by Theorem 16 is that, even though the gap period we constructed in the proof of Theorem 17 is information theoretically deterministic, it is not (computationally) known to the learner, i.e. the learner would never be able to (computationally) figure out when the sample n will arrive.*

We now consider a different scenario where we require the learner to be not only computable, but also has limited computational resources. We need the following notion. A *online learning scheme* is a triplet $(\mathcal{D}, \mathbf{R}, \Phi)$, where

1. $\mathcal{D} \in \{0,1\}^*$ is an *unlimited* database which the learner can use to store anything learned, initially an empty string;
2. Φ is the predictor which maps $\mathbb{N} \to \{0,1\}$, using the database $\mathcal{D}$ as an oracle;

3. **R** is the recorder that maps $\{0,1\}^* \times \mathbb{N} \times \{0,1\} \to \{0,1\}^*$, which updates the database $\mathcal{D}$ every time a new example and its label are revealed. Where needed, we use $\mathcal{D}_n$ to denote the state of the database after n instances and their labels have been revealed.

Definition 19 (Exact online learnable) *Let $\mathcal{H}$ be a class of functions from $\mathbb{N} \to \{0,1\}$. $\mathcal{H}$ is said to be exact online learnable, if there is an* online learning scheme $(\mathcal{D}, \boldsymbol{R}, \Phi)$ *such that for all $h \in \mathcal{H}$*

$$\sum_{n=1}^{\infty} 1\{\Phi^{\mathcal{D}_n}(n) \neq h(n)\} < \infty,$$

where $\mathcal{D}$ updates after every n,

$$\mathcal{D}_{n+1} = \boldsymbol{R}(\mathcal{D}_n, n, h(n)).$$

For any computable online learning scheme $(\mathcal{D}, \mathbf{R}, \Phi)$, we will specify **R** and Φ as algorithms. The time complexity of **R**, Φ and h is expressed in terms of $\log n$, i.e. the binary representation size of n. However, we use $\mathcal{D}$ as an oracle of **R** and Φ with no computational cost.

We focus on the worst case time complexity for **R** and Φ, *i.e.* over the most inconvenient database and function we are trying to learn. We say an online learning scheme runs in *uniformly exponential time*, if, no matter what database $\mathcal{D}$ is and what function h is in force, there exist some $c, N \in \mathbb{N}$ such that both **R** and Φ run in $\exp(c \log n) = n^c$ time for all $n \geq N$. We define *uniform polynomial time* similarly.

Theorem 20 *Let $\mathcal{H}$ be the class of all exponential time computable functions from $\mathbb{N} \to \{0,1\}$. Then there is no computable online learning scheme $(\mathcal{D}, \boldsymbol{R}, \Phi)$ that exactly learns $\mathcal{H}$ such that both **R** and Φ run in uniformly exponential time.*

We leave the proof of Theorem 20 to the appendix A. We made the following conjecture for polynomial time computable functions.

Conjecture 21 *Let $\mathcal{H}$ be the class of all polynomial time computable functions from $\mathbb{N} \to \{0,1\}$. Then there is no computable online learning scheme $(\mathcal{D}, \boldsymbol{R}, \Phi)$ that exactly learns $\mathcal{H}$ such that both **R** and Φ uniformly run in polynomial time.*

While we are unable to prove the conjecture, we can prove the following specialized version (see appendix A for a proof). We say the recorder **R** to be a naive recorder, if it simply appends $h(n)$ to the database $\mathcal{D}$ at the update of step n.

Theorem 22 *Let $\mathcal{H}$ be the class of all functions from $\mathbb{N} \to \{0,1\}$ whose time complexity is eventually bounded by $\log^{k+1} n$ time for some $k \geq 1$. Then there is no computable online learning scheme $(\mathcal{D}, \boldsymbol{R}, \Phi)$ that exactly learns $\mathcal{H}$ such that **R** is a naive recorder and Φ runs in uniformly $\log^k n$ time.*

We note the following interesting connection with time hierarchy theorem. Denote

$$\mathcal{H}_k = \{h : \exists \mathbf{TM} \; s.t. \; \forall n \in \mathbb{N}, \mathbf{TM}(n) = h(n) \text{ and } \mathbf{C}(\mathbf{TM}, n) \leq \log^k n\}.$$

Corollary 23 *Theorem 22 implies that $\mathcal{H}_{k+2} \backslash \mathcal{H}_k$ is not empty.*

Remark 24 *The time hierarchy theorem (Arora and Barak, 2009, Theorem 3.1) does not necessarily imply Theorem 22, since the predictor uses the database $\mathcal{D}$ as an zero-computational cost oracle when computing on n.*

6. Noisy labeling

In the previous sections we considered various setups in the eas-online learning paradigm when the labels are presented accurately. However, it is possible that the labels are corrupted by some noise. Surprisingly, we will show in this section that our eas-online learning paradigm are actually resistant to independent noises even if the sample are presented sequentially.

For simplicity, we will assume the domain to be $\mathcal{X} = \mathbb{N}$ and that the label is binary. We consider the following noisy process. At each time step i, we denote $\tilde{Z}_i = (X_i, h(X_i) \oplus R_n)$ to be the noisy sample-label pair, where R_n is a binary random variable with $\Pr(R_n = 1) \leq \eta$ and variables R_n are independent of the instances, labels, and in addition R_n are independent for each n.

We prove the following result.

Theorem 25 *Let $\mathcal{H}$ be a function class from $\mathbb{N} \to \{0, 1\}$, p is a distribution over $\mathbb{N}$. If $\mathcal{H}$ is effectively countable w.r.t. p and $\eta < \frac{1}{2}$, then there exists a learning strategy Φ, such that for all $h \in \mathcal{H}$ we have*

$$\Pr\left(\sum_{n=1}^{\infty} \ell(\Phi(\tilde{Z}_1^{n-1}, X_n), h(X_n)) < \infty\right) = 1,$$

where the randomness comes from both the sample and noise.

Proof Let $\mathcal{H}'$ a countable class that effectively covers $\mathcal{H}$ such that any two distinct functions in $\mathcal{H}'$ differs by a positive measure set in $\mathbb{N}$. Let $\mathcal{H}'_1 \subset \mathcal{H}'_2 \subset \cdots \mathcal{H}'$ be an nesting such that $|\mathcal{H}'_k| = k$ and $\bigcup_{k\in\mathbb{N}} \mathcal{H}'_k = \mathcal{H}'$.

The prediction happens in stages. At stage 0, we initialize by choosing an arbitrary function h_0 from $\mathcal{H}'$. Let h_{k-1} be the function we have after stage $k-1$. We will use h_{k-1} to make prediction in each stage k. At stage k we try to identify the underlying function as if it were in $\mathcal{H}'_k$. To do so, we note that each function in $\mathcal{H}'_k$ differs other functions by a positive measure subset of $\mathbb{N}$. One observes the samples sufficiently long so that there are m_k samples (instances may be repeated) in the difference sets of each pair of functions. We define $h_k \in \mathcal{H}'_k$ to be a function that has a smaller Hamming distance to the noisy labeling of samples at the difference positions relative to all other functions in $\mathcal{H}'_k$. If no such function exists, choose an arbitrary function in $\mathcal{H}'_k$.

We now show that the strategy indeed works. To do so, we choose $m_k = \frac{3 \log k}{2(0.5-\eta)^2}$. By the Hoeffding bound and union bound, with probability at least $1 - \frac{1}{k^2}$, we will find the correct underlying function at stage k, if it is in $\mathcal{H}'_k$. Now, since the underlying function must be in some $\mathcal{H}'_t$, the predictor must find it in finite steps. And by the Borel-Cantelli lemma, we will only miss it finitely many times with probability 1 since $\sum_{k\in\mathbb{N}} \frac{1}{k^2} < \infty$. After that we will make no errors. The theorem follows. ■

In the proof of Theorem 25, the exact probability $\Pr(R_n = 1)$ is not required to be known, only the upper bound η need be known. However, even the requirement of knowledge of η can be eliminated by a simple application of the doubling trick. Note that, Theorem 25 also holds if the instances are presented sequentially. In such a case, one observes for each instance exactly one noisy label. Therefore, there is no way estimate any single label (which would have been possible in the *i.i.d.* case). However, the proof of Theorem 25 shows that one would be able to identify the correct function with arbitrary high confidence by leveraging noisy labels of different instances.

7. Discussion

In this paper, we introduced an online learning paradigm where a learner is required to make only finitely many errors with probability one in an infinite horizon. The setup generalized the well known online learning setup of (Littlestone, 1988) to a more relaxed non-uniform consistency. We also demonstrated the relationship of our setup with the classic learning of recursive function literature as surveyed in (Zeugmann and Zilles, 2008), and showed that the randomness and resource restriction could bring richer picture when considering computability. We also investigated the case when noisy labels are presented, and showed that our setup are actually resistant to independent Bernoulli noise.

While our setup does not provide bounds (or rates) on how many errors will the learner be made, we should emphasize that we are also dealing with very rich classes that do not admit uniform consistency results. Our work is more focused on the conceptual foundation of learning and is meant to understand large scale problems that we have little prior knowledge in. Indeed, our approach also provide a different angle for a theorist to rethink "no free lunch" theorems. Philosophically, this is a question that underlies much of our scientific endeavor—can a physicist make only finite errors before finding a universal theory? Or will she perpetually move between models without every converging on one? See Cover (1973) and Appendix C for more discussions on this type of problems.

Acknowledgments

This work was supported by NSF grants CCF-1619452 and by the Center or Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

Jānis Martynovich Barzdiņš and RV Freivald. On the prediction of general recursive functions. In *Doklady Akademii Nauk*, volume 206, pages 521–524. Russian Academy of Sciences, 1972.

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

Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon van Handel, and Amir Yehudayoff. A theory of universal learning. *arXiv:2011.04483*, 2020.

Thomas M Cover. On determining the irrationality of the mean of a random variable. *The annals of Statistics*, 1(5):862–871, 1973.

Amir Dembo and Yuval Peres. A topological criterion for hypothesis testing. *The Annals of Statistics*, pages 106–117, 1994.

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

Sanjeev R Kulkarni and David N. C. Tse. A paradigm for class identification problems. *IEEE Transactions on Information Theory*, 40(3):696–705, 1994.

Amir Leshem. Cover's test of rationality revisited: Computability aspects of hypothesis testing. In *2006 IEEE 24th Convention of Electrical & Electronics Engineers in Israel*, pages 213–216. IEEE, 2006.

Lihong Li, Michael L Littman, Thomas J Walsh, and Alexander L Strehl. Knows what it knows: a framework for self-aware learning. *Machine learning*, 82(3):399–443, 2011.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Neri Merhav and Meir Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44 (6):2124–2147, 1998.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhanava Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

Narayana Santhanam and Venkat Anantharam. Agnostic insurability of model classes. *Journal of Machine Learning Research*, 16:2329–2355, 2015. URL http://jmlr.org/papers/v16/santhanam15a.html.

Amin Sayedi, Morteza Zadimoghaddam, and Avrim Blum. Trading off mistakes and don't-know predictions. In *Advances in Neural Information Processing Systems*, pages 2092–2100, 2010.

Dale Schuurmans. Characterizing rational versus exponential learning curves. *journal of computer and system sciences*, 55(1):140–160, 1997.

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

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

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

Changlong Wu and Narayana Santhanam. Being correct eventually almost surely. In *Information Theory (ISIT), 2019 IEEE International Symposium on*, pages 1989–1993. IEEE, 2019.

Thomas Zeugmann and Sandra Zilles. Learning recursive functions: A survey. *Theoretical Computer Science*, 397(1-3):4–56, 2008.

Chicheng Zhang and Kamalika Chaudhuri. The extended littlestones dimension for learning with mistakes and abstentions. In *Conference on Learning Theory*, pages 1584–1616, 2016.

Appendix A. Omitted proofs

Proof [Proof of Theorem 11] The sufficiency follows directly from Lemma 4. To prove the necessary condition, by the argument in the proof of Theorem 7, it is sufficient to show that there exists a countable dense subset of measurable functions from $\mathcal{X} \to \mathcal{Y}$ with the following metric

$$d(f, g) = \Pr_{x\sim\mu}(f(x) \neq g(x)).$$

To do so, we use a truncation argument. Wolog, we assume $\mathcal{Y} = \mathbb{N}$. We observe that there exists a class of countable measurable functions $\mathcal{P}_m$ that is dense in the measurable functions from $\mathcal{X} \to [m]$ for all $m \in \mathbb{N}$, since the σ-algebra on $\mathcal{X}$ is separable. We claim that $\bigcup_{m=1}^{\infty} \mathcal{P}_m$ is dense in the measurable functions from $\mathcal{X} \to \mathbb{N}$. Let $h : \mathcal{X} \to \mathbb{N}$ be an arbitrary measurable function and $\epsilon > 0$. There exists $m \in \mathbb{N}$ such that

$$\Pr_{x\sim\mu}(h(x) \geq m) \leq \epsilon/2.$$

Let h^m be the function such that $h^m(x) = h(x)$ if $h(x) \leq m$ and $h^m(x) = 1$ otherwise. We have $d(h^m, h) \leq \epsilon/2$. Now, choosing $h' \in \mathcal{H}_m$ with $d(h^m, h') \leq \epsilon/2$, we have $d(h, h') \leq \epsilon$. This completes the proof. ■

Proof [Proof of Theorem 20] We actually prove a stronger version of the theorem. Let $\mathcal{H}$ be the set of functions that can be computed within time n^{k+1} (hence exponential in $\log n$), we show that there is no online learning scheme $(\mathcal{D}, \mathbf{R}, \Phi)$ that exactly learns $\mathcal{H}$ such that $\mathbf{R}$ and Φ uniformly runs in $n^k/2$ time.

The proof uses a diagonalization argument. Assume to the contrary that we have such a scheme $(\mathcal{D}, \mathbf{R}, \Phi)$. We construct the following algorithm **ExpDiag**(n):

1. **Input**: n
2. **If** $n = 1$ **Output** 1.
3. Run scheme $(\mathcal{D}, \mathbf{R}, \Phi)$ on samples (k,**ExpDiag**(k)) with $k \leq n - 1$.
4. **Output** $1 - \Phi^{\mathcal{D}}(n)$.

To analyze the running time of **ExpDiag**(n), let $f(n)$ be the time needed to compute **ExpDiag** with input n. We have

$$f(n) = n^k + f(n - 1),$$

since $\mathbf{R}$ and Φ run in $n^k/2$ by assumption and one may reuse the database at the recursion steps. We have

$$f(n) \leq \sum_{i=1}^{n-1} i^k \leq n^{k+1}.$$

Therefore, the function that computed by **ExpDiag** is in $\mathcal{H}$. However, by construction **ExpDiag**$(n) \neq \Phi^{\mathcal{D}}(n)$ for all $n \geq 2$ which yields a contradiction as before. ■

Remark 26 *Note that similar argument cannot be generalized to the functions that computed in time $\log^{k+1} n$ against predictors runs in $\log^k n$ because establishing the database in step 3 will require $\Omega(n)$ steps in the naive way. However, we still believe this is true as in Conjecture 21. Theorem 22 establishes the conjecture when the database simply appends the true labels without processing.*

Proof [Proof of Theorem 22] We use a standard approach that reduces the polynomial to exponential case. Let $\mathcal{H}$ be the class of all functions that can be computed in time $\log^{k+1} n$. Suppose to the contrary, $(\mathcal{D}, \mathbf{R}, \Phi)$ is a scheme that exactly learns $\mathcal{H}$ such that $\mathbf{R}$ is naive recorder and Φ runs in $\log^k n$. For any function h' that can be computed in n^{k+1} time, we construct a function h that can be computed in $\log^{k+1} n$ time as follows

$$h(n) = \begin{cases} h'(t), \text{ if } n = 2^t \text{ for some } t \in \mathbb{N} \\ 1, \text{ otherwise} \end{cases}.$$

Consider the following predictor Φ':

1. **Input**: n and naive recording $\mathcal{D}$ of h' upto $n-1$
2. Simulate Φ with input 2^n as follows: if Φ queries database at position T such that $T = 2^t$ we query the tth position in $\mathcal{D}$. In all other positions, we know h took value 1.
3. **Output** $\Phi^{\mathcal{D}}(2^n)$.

By assumption Φ makes finitely many errors on h, thus Φ' makes finitely errors on h' as well. Clearly, we have Φ' runs in $(\log 2^n)^k = n^k$ time and it exactly learns the class of functions that can be computed in n^{k+1}. This contradicts Theorem 20. ■

Proof [Proof of Corollary 23] Suppose not. We will construct an online learning scheme $(\mathcal{D}, \mathbf{R}, \Phi)$ that exact learns $\mathcal{H}_{k+2}$ such that $\mathbf{R}$ is naive recorder and Φ runs in $\log^{k+1} n$, which will contradict Theorem 22.

To do so, we enumerate all Turing machines $\mathbf{TM}_1, \mathbf{TM}_2, \cdots$. The predictor Φ maintains two indices t, i. At each time step n, both t and i are initialized to 1. The predictor *simulates* $\log^k i$ instructions on $\mathbf{TM}_t(i)$. If it stops within those $\log^k i$ instructions and its output matches $h(i)$, keep $t = t$ and increment $i = i + 1$. Else, move to the next machine, $t = t + 1$ and increment $i = i + 1$.

This process continues for $\frac{1}{2}\log^{k+1} n$ net time (this includes the time taken to simulate steps on all Turing machines thus far, and all relevant overheads, including that for incrementing indices). Then set $i = n$ and simulate $\mathbf{TM}_t(n)$ till the run for a net $\log^{k+1}(n)$ time. If $\mathbf{TM}_t$ stops by then, the predictor outputs $\mathbf{TM}_t(n)$, otherwise it outputs 1.

We show that this scheme is an exact online learning scheme for $\mathcal{H}_{k+2}$. For any $h \in \mathcal{H}_{k+2}$, we have an Turing machine $\mathbf{TM}_j$ that outputs $h(n)$ for all n within $\log^k n$ time, since $\mathcal{H}_{k+2} = \mathcal{H}_k$ by assumption. Note that t increases iff the predictor makes errors. Since in addition, $\frac{1}{2}\log^{k+1} n \to \infty$, we know that t will hit j eventually. Once t hits j, note that we never increment it since $\mathbf{TM}_j(i)$ outputs $h(i)$ within $\log^k i$ time for all i. If the time runs out before we complete the simulation of $\mathbf{TM}_j$, note again that t is not incremented. Finally, since the overhead on universal Turing machine is a $\log \log n$ factor on time complexity (Arora and Barak, 2009, Theorem 1.13), we know that for large enough n, the algorithm above actually completes the simulation of $\mathbf{TM}_j(n)$. ■

Appendix B. Non-uniform time complexity

If we relax the requirement of worst case time complexity, we show that polynomial time computable functions can be learned exactly in (pointwise) polynomial time.

Theorem 27 *Let $\mathcal{H}$ be the class of all polynomial time computable functions from $\mathbb{N} \to \{0,1\}$. Then there exists a computable exact online learning scheme $(\mathcal{D}, \boldsymbol{R}, \Phi)$ that exactly learns $\mathcal{H}$ such that both $\boldsymbol{R}$ and Φ run in point-wise polynomial time.*

Proof The database $\mathcal{D}$ consists of a triplet $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2, k)$ where $\mathcal{D}_1, \mathcal{D}_2$ encodes for Turing machines and $k \in \mathbb{N}$. The predictor Φ works as follows:

1. **Input**: n with database $\mathcal{D}$
2. **If** $\mathcal{D}_1$ is not a valid encoding of a Turing machine, **Output** 1.
3. Simulate the Turing machine encoded by $\mathcal{D}_1$ on input n for $\log^k n$ steps.
4. **If** $\mathcal{D}_1$ stops in $\leq \log^k n$ steps **Output** $\mathcal{D}_1(n)$. **Else**, **Output** 1.

The recorder **R** works as follows:

1. **Input**: Old database $\mathcal{D}$, prediction $\Phi^{\mathcal{D}}(n)$ and $n, h(n)$
2. **Output**: New database $\mathcal{D}$
3. **If** $n = 1$, set $\mathcal{D} = \{\phi, \phi\}$, where ϕ is empty string.
4. **If** $\Phi^{\mathcal{D}}(n) = h(n)$, do not change on $\mathcal{D}$ and **Return**.
5. **Else**:
   a. **If** $\mathcal{D}_1 \neq \mathcal{D}_2$, **Set** $\mathcal{D}_1 =$ **NEXT**$(\mathcal{D}_1)$, where **NEXT** maps strings in $\{0,1\}^*$ to the next string in alphabet order, and $k = k + 1$
   b. **Else**, **Set** $\mathcal{D}_1 = \phi$ and $\mathcal{D}_2 =$ **NEXT**$(\mathcal{D}_2)$.

Suppose that the underlying function $h \in \mathcal{H}$ can be computed in $O(\log^t n)$ time by some Turing machine with encoding $\mathcal{D}_h$. By the construction of **R**, the database $\mathcal{D}$ changes iff the prediction made by Φ is wrong.

We show that the database changes only finitely many times, thus proving that the total number of errors is finite. If the database changes infinitely many times, note that it will also hit $\mathcal{D}_h$ infinitely many times. $\mathcal{D}_h$ is simulated for $\log^k n$ steps in Step 3. of the predictor, where k increases every time there is an error. Therefore, k will eventually be large enough that $\mathcal{D}_h$ is simulated for enough steps that it halts and outputs the correct value of $h(n)$. Following this, the database can not change, contradicting the assumption that the database changes infinitely many times. ■

Appendix C. More variations

Contrast to the finite error guarantee, perhaps a more natural non-uniform consistency of online learning one may think is to let the average loss per time step convergence to zero. More formally, one may wish to find a predictor such that

$$\Pr\left(\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \ell(\Phi(Z_1^{n-1}, X_n), h(X_n)) = 0\right) = 1.$$

However, such a guarantee might not be attractive if one only considers the consistency. Since we can show that the class of all measurable functions from $\mathbb{R} \to \{0,1\}$ are actually learnable in that sense.

Theorem 28 *Let $\mathcal{H}$ be the class of all measurable functions from $\mathbb{R} \to \{0,1\}$, μ is an arbitrary distribution over $\mathbb{R}$ that is unknown to the learner, ℓ is the classification loss. Then there exist a learning strategy Φ such that for all $h \in \mathcal{H}$ we have*

$$\Pr\left(\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} \ell(\Phi(Z_1^{n-1}, X_n), h(X_n)) = 0\right) = 1,$$

where Z_i and X_i are defined as in Section 2.

Proof [Sketch of Proof] Let $\mathcal{G}$ be a countable class of function from $\mathbb{R} \to \{0,1\}$ that is dense in $\mathcal{H}$. Note that, since the Borel σ-algebra is separable, we known such a class exist and independent of the underlying distribution. The prediction partitioned into stages. At stage k, the learner tries to identify a function h_k in $\mathcal{G}$ that is ϵ_k-close to the underlying function with confidence at least $1 - \frac{1}{k^2}$. This can be easily achieved by using a structural risk minimization argument. Note that we will use h_{k-1} to make the prediction at stage k, and use the sample observed at stage k to find h_k for the next stage. We now choose ϵ_k small enough so that the probability that it has average error jump above ϵ_k in the infinite horizon is at most $\frac{1}{k^2}$. This can be done by Chernoff bound with a union bound, and observe that $\sum_{n=1}^{\infty} \exp(-2n\epsilon_k)$ convergence and goes to zero when ϵ_k goes to zero. The theorem now follows by Borel-Cantelli lemma, since $\frac{1}{k^2}$ is summable. ■

One may also consider the scenario when the rate of convergence to zero is controlled. It can be shown that if the class has finite VC-dimension, then we can have rate of $O(\frac{\log n}{n})$ in expectation, see Haussler et al. (1994). However, this does not automatically give us an almost sure upper bound on the cumulative errors, since the errors at different time steps are *correlated*. We now show that we can indeed achieve a $O(\frac{\log^2 n}{n})$ rate almost surely. To do so, we use a doubling trick. We partition the prediction into stages. At stage k, we will see 2^k sample-label pairs, and we generate a hypothesis h_k using the observed pairs. We then use h_k to make predictions for another 2^k steps, at the end of which we move to stage $k+1$. To see why this approach works, we denote I_n to be the indicator that an error occurred at step n. By construction, we know that the indicators in each phase are *independent*. We can therefore employ multiplicative Chernoff bound to show that with probability at least $1 - O(1/2^k)$ the number of errors at stage k is upper bounded by $O(k)$ (using empirical risk minimization to obtain h_k is sufficient to achieve this). Therefore, by the Borel-Cantelli lemma, the partial sums of the indicators $\{I_n\}$ up to stage k will be eventually upper bounded by $O(k^2)$ with probability 1. Since the first n steps will cover at most $\log(n)$ stages, the result follows. Note that, the $O(\log^2 n/n)$ rate is not meant to be optimal, it is not hard to see that similar argument could establish a $O(\log n \log\log n/n)$ rate almost surely by replacing the h_k with an optimal predictor that has errors $O(\log k/2^k)$ with probability $1 - O(1/k^2)$. We leave it as an open problem to obtain the optimal almost sure rate with finite VC-dimension.

A binary hypothesis class $\mathcal{H}$ is said to be *closed*, if for any measurable function f and distribution μ we have

$$\inf_{h\in\mathcal{H}} \Pr_{x\sim\mu}[h(x) \neq f(x)] = 0,$$

then $f \in \mathcal{H}$. Recently, Bousquet et al. (2020) showed that for *closed* hypothesis classes the expected rate (i.e. $\mathbb{E}[I_n]$) can either be exponential or linear or arbitrary slow (here we have used the closeness notion in replacing of the adversary setting in Bousquet et al. (2020)). Note that the closeness of the class is a crucial part to establish their lower bounds. For example, consider the class $\mathcal{R}$ of all linear threshold functions over $[0, 1]$ with rational parameter, we know that the class can be learned with finitely many errors almost surely. However, $\mathcal{R}$ is not closed and it has an infinite Littlestone tree, the result of Bousquet et al. (2020) only implies that the *closure* of $\mathcal{R}$ has $\mathbb{E}[I_n] = \Omega(1/n)$. It is therefore an interesting problem to investigate whether similar phenomenon will happen for general classes (i.e. not necessarily closed) and in the almost sure scenario.