Yale University CPSC 683
Stability in Machine Learning: Generalization, Privacy & Replicability

Lecture 10: A Theory of Learning Curves

Instructor: Alkis Kalavasis, alkis.kalavasis@yale.edu

Lecture 10 presents a setting that classical VC theory fails to explain. We will be interested in *learning curves*. This problem was studied by Bousquet et al. (2021). At a technical level, stability will be crucial in order to obtain one of the key results of the paper.

1 A Theory of Learning Curves

How quickly does the error of a learning algorithm drop for a given class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ of concepts as the number of samples increases?

The standard way to measure the performance of a learning algorithm is by plotting its *learning curve*, i.e., by plotting the decay of the error rate as a function of the number n of training examples. This is also standard in *scaling laws*, when one compares performance by scaling some resource such as training dataset size or model size.

A natural first question is to wonder whether the classical theory of learning, the PAC model of Vapnik-Chervonenkis and Valiant, could be used to explain learning curves. The answer is *no*.

The reason is that the PAC model adopts a *minimax* perspective. Let us denote by $\mathrm{RE}(H)$ the family of distributions P for which the concept class H is realizable, i.e., $\inf_{h\in H} \mathbf{E}_P[\mathrm{er}(h)] = 0$. The fundamental result of PAC learning theory states that

$$\inf_{\hat{h}_n} \sup_{P\in\mathrm{RE}(H)} \mathbf{E}[\mathrm{er}(\hat{h}_n)] \asymp \min\left(\frac{\mathrm{vc}(H)}{n}, 1\right),$$

where $\mathrm{vc}(H)$ is the VC dimension of H.

In other words, PAC learning theory is concerned with the best *worst-case* error over all realizable distributions, that can be achieved by means of a learning algorithm $\hat{h}_n$.

PAC model can only bound the upper envelope of the learning curves over all possible data distributions. Observe that the worst-case distribution can change with the sample size n! However, this is not reflected in real world, where the data source is typically fixed in any given scenario. The learner may choose the number of training examples to use but, as the sample increases, the data source remains the same.

VC theory immediately implies a fundamental *dichotomy* of uniform rates: every concept class H has a uniform rate that is either *linear* $\frac{c}{n}$ or *bounded away from zero*, depending on the finiteness of the combinatorial parameter $\mathrm{vc}(H)$. The term *uniform* comes from the fact that it holds uniformly over all distributions R.

2 Trichotomy of Possible Universal Rates

Bousquet et al. (2021) propose a model for learning curves, called the *universal rates* models. The term *universal* is important and replaces the term uniform: a given property of the learner (such as consistency or rate) should hold for *every* realizable distribution P, but *not* uniformly over all distributions. For example, a class H is universally learnable at rate R if the following holds:

$$\exists\, \hat{h}_n \quad \text{s.t.} \quad \forall\, P \in \mathrm{RE}(H), \quad \exists\, C, c > 0 \quad \text{s.t.} \quad \mathbf{E}[\mathrm{er}(\hat{h}_n)] \le CR(cn) \text{ for all } n.$$

The crucial difference between universal rates and the uniform PAC setting is that *the constants C, c are allowed to depend on P*: thus universal learning is able to capture *distribution-dependent* learning curves for a given learning problem.

Question. Given a class H, what is the fastest rate at which H can be universally learned?

We will need the following definitions.

Definition 1: Learning Rates

Let H be a concept class, and let $R : \mathbb{N} \to [0, 1]$ with $R(n) \to 0$ be a rate function.

- H is learnable at rate R if there is a learning algorithm $\hat{h}_n$ such that for every realizable distribution P, there exist $C, c > 0$ for which $\mathbf{E}[\mathrm{er}(\hat{h}_n)] \leq CR(cn)$ for all n.
- H is not learnable at rate faster than R if for every learning algorithm $\hat{h}_n$, there exists a realizable distribution P and $C, c > 0$ for which $\mathbf{E}[\mathrm{er}(\hat{h}_n)] \geq CR(cn)$ for infinitely many n.
- H is learnable with optimal rate R if H is learnable at rate R and H is not learnable faster than R.
- H requires arbitrarily slow rates if, for every $R(n) \to 0$, H is not learnable faster than R.

The main result of Bousquet et al. (2021) is a remarkable **trichotomy**: there are only three possible rates of universal learning. More precisely, we show that the learning curves of any given concept class decay either at an exponential, linear, or arbitrarily slow rates. Moreover, each of these cases is completely characterized by appropriate combinatorial parameters, and we exhibit optimal learning algorithms that achieve the best possible rate in each case.

Theorem 1: Trichotomy of Universal Learning Rates

For every concept class H with $|H| \geq 3$, exactly one of the following holds.

- H is learnable with optimal rate e^{-n}.
- H is learnable with optimal rate $\frac{1}{n}$.
- H requires arbitrarily slow rates.

The condition $|H| \geq 3$ avoids some trivially learnable classes.

More to that, one can provide combinatorial measures to characterize when each rate occurs.

Theorem 2: Combinatorial Characterization of Universal Rates

For every concept class H with $|H| \geq 3$, the following hold:

- If H does not have an infinite Littlestone tree, then H is learnable with optimal rate e^{-n}.
- If H has an infinite Littlestone tree but does not have an infinite VCL tree, then H is learnable with optimal rate $\frac{1}{n}$.
- If H has an infinite VCL tree, then H requires arbitrarily slow rates.

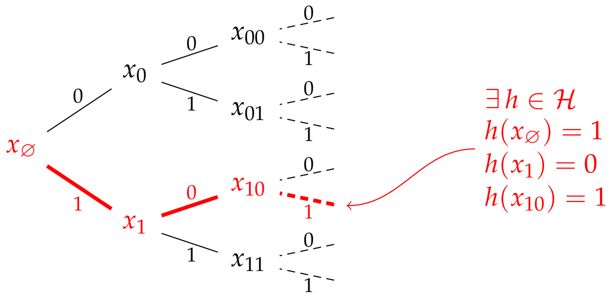


Figure 1: Figure taken from Bousquet et al. (2021). A Littlestone tree of depth 3. Every branch is consistent with a concept $h \in \mathcal{H}$. This is illustrated here for one of the branches. From Bousquet et al. (2021).

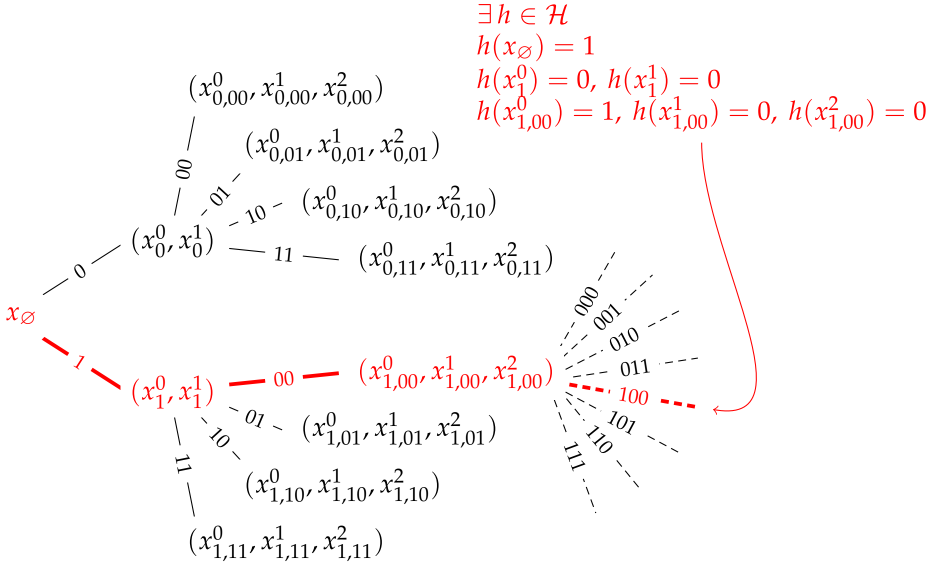

$\exists\, h \in \mathcal{H}$
$h(x_\varnothing) = 1$
$h(x_1^0) = 0,\ h(x_1^1) = 0$
$h(x_{1,00}^0) = 1,\ h(x_{1,00}^1) = 0,\ h(x_{1,00}^2) = 0$
$x_\varnothing$
0
1
(x_0^0, x_0^1)
(x_1^0, x_1^1)
00
01
10
11
$(x_{0,00}^0, x_{0,00}^1, x_{0,00}^2)$
$(x_{0,01}^0, x_{0,01}^1, x_{0,01}^2)$
$(x_{0,10}^0, x_{0,10}^1, x_{0,10}^2)$
$(x_{0,11}^0, x_{0,11}^1, x_{0,11}^2)$
00
01
10
11
$(x_{1,00}^0, x_{1,00}^1, x_{1,00}^2)$
$(x_{1,01}^0, x_{1,01}^1, x_{1,01}^2)$
$(x_{1,10}^0, x_{1,10}^1, x_{1,10}^2)$
$(x_{1,11}^0, x_{1,11}^1, x_{1,11}^2)$
000
001
010
011
100
101
110
111

Figure 2: Figure taken from Bousquet et al. (2021). A VCL tree of depth 3. Every branch is consistent with a concept $h \in \mathcal{H}$. This is illustrated here for one of the branches.

3 The Exponential Rates Proof - Stability is the Key

The proof of the exponential rates result is pretty surprising and the route one has to cross the get it is highly non-trivial and novel. Stability will be a key to that result.

3.1 The Online Learning Game

Let $\mathcal{X}$ be a domain, and let the concept class $\mathcal{H}$ be a collection of functions $h : \mathcal{X} \to \{0,1\}$. We consider a two-player game between the *learner* P_L and an *adversary* P_A. The game is played in rounds. In each round $t \geq 1$:

- P_A chooses a point $x_t \in \mathcal{X}$.
- P_L predicts a label $\hat{y}_t \in \{0,1\}$ for that point.
- P_A reveals the true label $y_t = h(x_t)$ for some function $h \in \mathcal{H}$ that is consistent with the previous label assignments $h(x_1) = y_1, \ldots, h(x_{t-1}) = y_{t-1}$.

The learner makes a mistake in round t if $\hat{y}_t \neq y_t$. The goal of the learner is to make as few mistakes as possible and the goal of the adversary is to cause as many mistakes as possible.

The adversary need not choose a target concept $h \in H$ in advance, but must ensure that the sequence $\{(x_t, y_t)\}_{t=1}^{\infty}$ is *realizable* by H in the sense that for all $T \in \mathbb{N}$ there exists $h \in H$ such that $h(x_t) = y_t$ for all $t \leq T$.

We will say that $\mathcal{H}$ is online learnable if there is a strategy

$$\hat{y}_t = \hat{y}_t(x_1, y_1, \ldots, x_{t-1}, y_{t-1}, x_t),$$

that makes only finitely many mistakes, regardless of what realizable sequence $\{(x_t, y_t)\}_{t=1}^{\infty}$ is presented by the adversary.

Based on what we have already seen in the previous lectures, this game is quite similar to how we derived the Littlestone dimension. However there is a crucial catch. Here we ask only that the strategy makes a finite number of mistakes on any input, without placing a uniform bound on the number of mistakes.

> **Theorem 3**
>
> For any concept class $\mathcal{H}$, we have the following dichotomy.
>
> - If $\mathcal{H}$ does not have an infinite Littlestone tree, then there is a strategy for the learner that makes only finitely many mistakes against any adversary.
> - If $\mathcal{H}$ has an infinite Littlestone tree, then there is a strategy for the adversary that forces any learner to make a mistake in every round.
>
> In particular, $\mathcal{H}$ is online learnable if and only if it has no infinite Littlestone tree.

Question: Is an infinite Littlestone tree and an infinite Littlestone dimension the same? (Think).

3.2 A Gale-Stewart Game

The first surprising step of the proof is the following connection. We view the online learning game from a new perspective that fits to classical game theory. However, classical game theory usually deals with finite games. Here we are playing an *infinite two-player game.*

From a game-theoretic perspective, we can ask: How should an adversary force mistakes to the learning player?

For $x_1, \ldots, x_t \in \mathcal{X}$ and $y_1, \ldots, y_t \in \{0,1\}$, consider the class

$$\mathcal{H}_{x_1,y_1,\ldots,x_t,y_t} := \{h \in \mathcal{H} : h(x_1) = y_1, \ldots, h(x_t) = y_t\}$$

of hypotheses that are consistent with the sequence of moves $x_1, y_1, \ldots, x_t, y_t$. An adversary P_A who aims to maximize the number of mistakes the learner makes will pick a sequence of x_t, y_t with $y_t \neq \hat{y}_t$ for as many rounds in a row as possible.

Using the above notation, the adversary tries to keep $\mathcal{H}_{x_1,1-\hat{y}_1,\ldots,x_t,1-\hat{y}_t} \neq \varnothing$ as long as possible. When this set would become empty (for every possible x_t), however, the only consistent choice of label is $y_t = \hat{y}_t$, so the learner makes no mistakes from that point onwards.

We can define the following game $\mathfrak{G}$. There are two players P_A and P_L. In each round τ:

- Player P_A chooses a point $\xi_\tau \in \mathcal{X}$ and shows it to Player P_L.
- Then, Player P_L chooses a point $\eta_\tau \in \{0,1\}$.

Player P_L wins the game in round τ if $\mathcal{H}_{\xi_1,\eta_1,\ldots,\xi_\tau,\eta_\tau} = \varnothing$. Player P_A wins the game if the game continues indefinitely. In other words, the set of winning sequences for P_L is

$$\mathsf{W} = \{(\boldsymbol{\xi}, \boldsymbol{\eta}) \in (\mathcal{X} \times \{0,1\})^\infty : \mathcal{H}_{\xi_1,\eta_1,\ldots,\xi_\tau,\eta_\tau} = \varnothing \text{ for some } 0 \leq \tau < \infty\}$$

This set of sequences W is *finitely decidable* in the sense that the membership of $(\boldsymbol{\xi}, \boldsymbol{\eta})$ in W is witnessed by a *finite subsequence*. Games that satisfy this property are called *Gale-Stewart (GS) games*. These games come with a fundamental property:

exactly one of P_A or P_L has a winning strategy in this game.

Let us now go back to the online game of the previous section. The game $\mathfrak{G}$ is fundamentally related to Littlestone trees:

P_A has a winning strategy in the Gale-Stewart game $\mathfrak{G}$ $\iff$ $\mathcal{H}$ has an infinite Littlestone tree.

This is interesting but how is it helpful to our goal? The main challenge in the proof is constructing a learning algorithm that achieves exponential rate for every realizable P. However, up to now, we only deal with the online setting where there is no distribution P.

Let us assume in the remainder that $\mathcal{H}$ has no infinite Littlestone tree. Winning strategies of Gale-Stewart games for P_L essentially yield the existence of a sequence of universally measurable functions $\hat{Y}_t : (\mathcal{X} \times \{0,1\})^{t-1} \times \mathcal{X} \to \{0,1\}$ that solve the online learning problem. We will use these winning strategies as classifiers for the statistical setting.

Define the data-dependent classifier

$$\hat{y}_{t-1}(x) := \hat{Y}_t(X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, x).$$

The first observation is that this online algorithm is also applicable in the statistical setting. In particular, one can show that

$$\mathbf{P}\{\mathrm{er}(\hat{y}_t) > 0\} \to 0 \text{ as } t \to \infty.$$

This certainly shows that $\mathbf{E}[\mathrm{er}(\hat{y}_t)] \to 0$ as $t \to \infty$. Thus the online learning algorithm yields a consistent algorithm in the statistical setting. This is how we used stability (i.e., online learnability) to get a consistent learning algorithm in the probabilistic setting.

This, however, does not yield any bound on the learning rate. It could be the case that the error of $\hat{y}_t$ goes to 0 quite slowly.

The idea now is to *boost* the rate of this algorithm. We will build a new algorithm on the basis of $\hat{y}_t$ that guarantees an exponential learning rate.

As a first observation, suppose we knew a number t^* so that $\mathbf{P}\{\mathrm{er}(\hat{y}_{t^*}) > 0\} < \frac{1}{4}$. This t^* exists and is finite (why?).

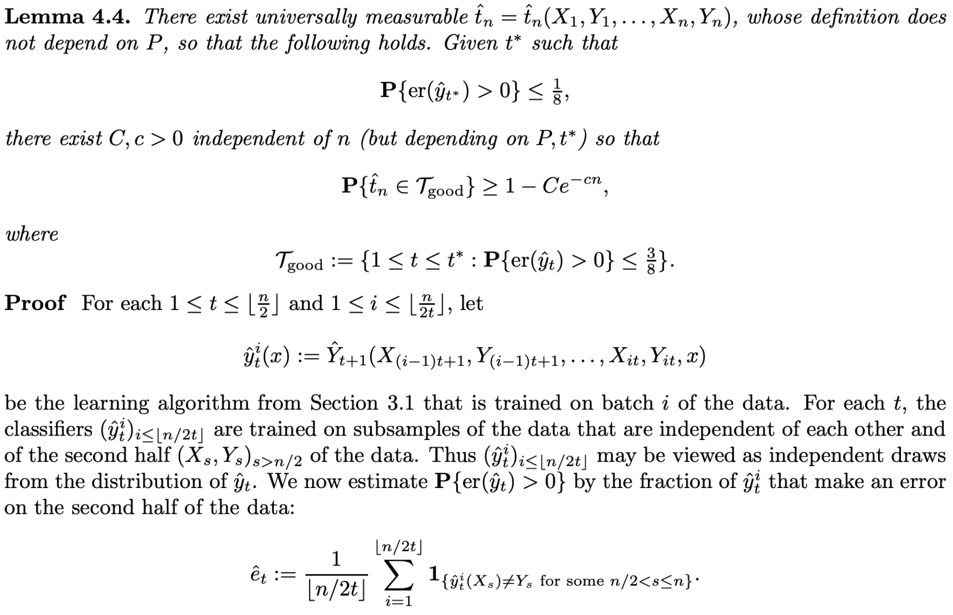
Then we could output a classifier $\hat{h}_n$ with exponential rate; the idea is just classical boosting argument.

- Break up the data $X_1, Y_1, \ldots, X_n, Y_n$ into $\lfloor n/t^* \rfloor$ batches, each of length t^*.
- Train a classifier $\hat{y}_{t^*}$ separately for each batch.
- Choose $\hat{h}_n$ to be the majority vote among these classifiers.

Now, by the definition of t^* and Hoeffding's inequality, the probability that more than $1/3$ of the classifiers has positive error is exponentially small! It follows that the majority vote $\hat{h}_n$ errs only with exponentially small probability, meaning that its learning curve drops exponentially fast. Note that this t^* depends on P and so the constants in the rate depend on P.

This last comment is exactly the problem with this idea: t^* depends on the unknown distribution P, so we cannot assume it is known to the learner. However, there is a smart fix to that: first, we will construct an estimate $\hat{t}_n$ for t^* from the data; and then we apply the above majority algorithm with batch size $\hat{t}_n$ instead t^*.

The estimation of t^* can be done using samples as follows:



Lemma 4.4. *There exist universally measurable $\hat{t}_n = \hat{t}_n(X_1, Y_1, \ldots, X_n, Y_n)$, whose definition does not depend on P, so that the following holds. Given t^* such that*

$$\mathbf{P}\{\mathrm{er}(\hat{y}_{t^*}) > 0\} \le \tfrac{1}{8},$$

there exist $C, c > 0$ independent of n (but depending on P, t^*) so that

$$\mathbf{P}\{\hat{t}_n \in \mathcal{T}_{\mathrm{good}}\} \ge 1 - Ce^{-cn},$$

where

$$\mathcal{T}_{\mathrm{good}} := \{1 \le t \le t^* : \mathbf{P}\{\mathrm{er}(\hat{y}_t) > 0\} \le \tfrac{3}{8}\}.$$

Proof For each $1 \le t \le \lfloor \frac{n}{2} \rfloor$ and $1 \le i \le \lfloor \frac{n}{2t} \rfloor$, let

$$\hat{y}_t^i(x) := \hat{Y}_{t+1}(X_{(i-1)t+1}, Y_{(i-1)t+1}, \ldots, X_{it}, Y_{it}, x)$$

be the learning algorithm from Section 3.1 that is trained on batch i of the data. For each t, the classifiers $(\hat{y}_t^i)_{i \le \lfloor n/2t \rfloor}$ are trained on subsamples of the data that are independent of each other and of the second half $(X_s, Y_s)_{s > n/2}$ of the data. Thus $(\hat{y}_t^i)_{i \le \lfloor n/2t \rfloor}$ may be viewed as independent draws from the distribution of $\hat{y}_t$. We now estimate $\mathbf{P}\{\mathrm{er}(\hat{y}_t) > 0\}$ by the fraction of $\hat{y}_t^i$ that make an error on the second half of the data:

$$\hat{e}_t := \frac{1}{\lfloor n/2t \rfloor} \sum_{i=1}^{\lfloor n/2t \rfloor} \mathbf{1}_{\{\hat{y}_t^i(X_s) \neq Y_s \text{ for some } n/2 < s \le n\}}.$$

Figure 3: Taken from Bousquet et al. (2021). Estimation of the critical time where the algorithms stops making mistakes with some constant probability $> 1/2$.

4 The Landscape of Universal Rates

In this section we provide further examples. The main aim of this section is to illustrate important distinctions with the uniform setting and other basic concepts in learning theory, which are illustrated schematically in Figure 4.

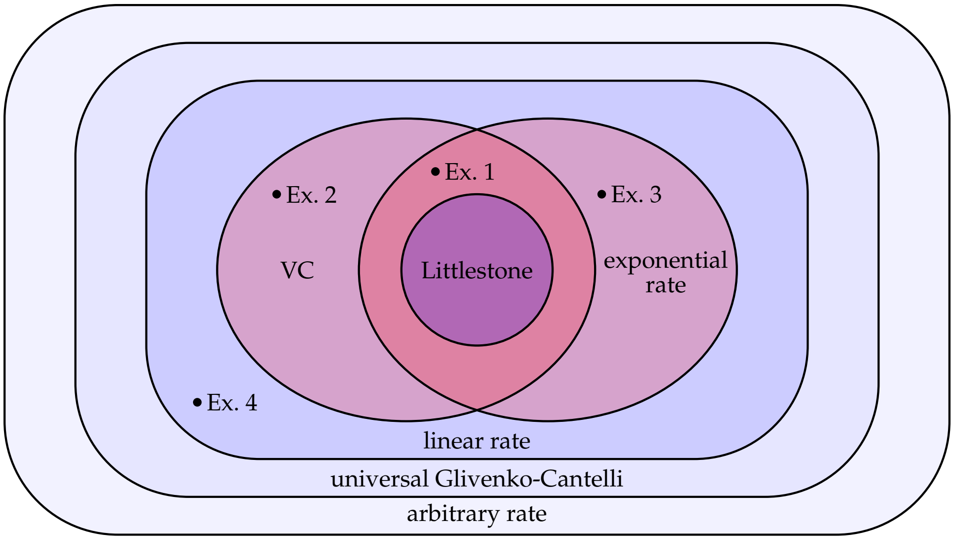


Figure 4: Figure taken from Bousquet et al. (2021). A Venn diagram depicting the trichotomy and its relation with uniform and universal learnability. While the focus here is on statistical learning, note that this diagram also captures the distinction between uniform and universal online learning.

We begin by giving four examples that illustrate that the classical PAC learning model (which is characterized by finite VC dimension) is not comparable to the universal learning model.

Example 1: VC with exponential rate

Consider the class $\mathcal{H} \subseteq \{0,1\}^{\mathbb{N}}$ of all threshold functions $h_t(x) = \mathbf{1}_{x \geq t}$ where $t \in \mathbb{N}$.

1. It is a VC class with VC dimension 1.
2. It is learnable at an exponential rate since it does not have an infinite Littlestone tree.
3. It has unbounded Littlestone dimension (it shatters Littlestone trees of arbitrary finite depths), so that it does not admit an online learning algorithm that makes a uniformly bounded number of mistakes.

Example 2: VC with linear rate

Consider the class $\mathcal{H} \subseteq \{0,1\}^{\mathbb{R}}$ of all threshold functions $h_t(x) = \mathbf{1}_{x \geq t}$, where $t \in \mathbb{R}$.

1. It is a VC class with VC dimension 1.
2. It is not learnable at an exponential rate (it has an infinite Littlestone tree).
3. The optimal rate is linear.

Example 3: Exponential rate but not VC

Let $\mathcal{X} = \bigcup_k \mathcal{X}_k$ be the disjoint union of finite sets $|\mathcal{X}_k| = k$. For each k, let $\mathcal{H}_k = \{\mathbf{1}_S : S \subseteq \mathcal{X}_k\}$, and consider the concept class $\mathcal{H} = \bigcup_k \mathcal{H}_k$.

1. This class has an unbounded VC dimension.

2. It is universally learnable at an exponential rate. To establish the latter, it suffices to prove that $\mathcal{H}$ does not have an infinite Littlestone tree. Indeed, once we fix any root label $x \in \mathcal{X}_k$ of a Littlestone tree, only $h \in \mathcal{H}_k$ can satisfy $h(x) = 1$, and so the hypotheses consistent with the subtree corresponding to $h(x) = 1$ form a finite class. This subtree can therefore have only finitely many leaves, contradicting the existence of an infinite Littlestone tree.

Example 4: Linear rate but not VC

Assume that $\mathcal{X}$ is the disjoint union of $\mathbb{R}$ and finite sets $\mathcal{X}_k$ with $|\mathcal{X}_k| = k$, and $\mathcal{H}$ is the union of the class of all threshold functions on $\mathbb{R}$ and the classes $\mathcal{H}_k = \{\mathbf{1}_S : S \subseteq \mathcal{X}_k\}$.

1. This class has an unbounded VC dimension.
2. It is universally learnable at a linear rate. To establish the latter, it suffices to note that $\mathcal{H}$ has an infinite Littlestone tree, but $\mathcal{H}$ cannot have an infinite VCL tree. Indeed, once we fix any root label $x \in \mathcal{X}$, the class $\{h \in \mathcal{H} : h(x) = 1\}$ has finite VC dimension, and thus the corresponding subtree of the VCL tree must be finite.

References

Bousquet, O., Hanneke, S., Moran, S., van Handel, R., and Yehudayoff, A. (2021). A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 532–541, New York, NY, USA. Association for Computing Machinery.