Sample Compression Schemes

Alireza Kazemipour

July 29, 2025

1 Introduction

"Learning and compression are known to be deeply related to each other. Learning procedures perform compression, and compression is an evidence of and is useful in learning." [5] One important topic in studying the theory of learning is PAC Learnability where a learner tries to learn an unknown labelling binary function out of a hypothesis class through having access to a subsample of an unknown distribution of data. The learner is obligated to learn in polynomial time, polynomial with respect to the inverse of the approximation error and inverse of the confidence bounds. While it's proven that every finite hypothesis class is PAC learnable [7], this is a sufficient condition for PAC learnability and, [1] showed that there are hypothesis classes with infinite size that are PAC learnable and expressed that the necessary and sufficient condition for PAC learnability is the finiteness of the VC dimension of the hypothesis class not its size.

In this project we try to demonstrate that the existence of a suitable data compression scheme is sufficient to ensure PAC learnability [4].This approach provides an alternative to that of **(author?)** [1]'s, which used the VC dimension to classify PAC learnable concepts. The bounds are derived directly from what we will call the kernel size of the algorithms rather than from the VC dimension of the hypothesis class and they provide weaker conditions for PAC learnability.

2 Settings

2.1 Formal model

In Probably Approximately Correct (PAC) learning, the data comes from a domain set $\mathcal{X}$, there is an unknown probability distribution over the domain set $\mathcal{D} : \mathcal{X} \to [0, 1]$, labels are binary $\mathcal{Y} = \{0, 1\}$ and, the true labeling function is unknown $f : \mathcal{X} \to \mathcal{Y}$. The learner has access to an independently identically distributed (i.i.d) dataset $\mathcal{S}^m = ((x_1, y_1), \cdots, (x_m, y_m))$ where $x_i \sim \mathcal{X}$ and $y_i = f(x_i)$ and the learner's output is $h \in \mathcal{H}$, h: $\mathcal{X} \to \mathcal{Y}$ where $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is a concept class that agent choose to learn function from in order to estimate the unknown f and the measure of success is the expected true loss of the learned concept:

$$L_{\mathcal{D},f}(h) := \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)] := \mathcal{D}(\{x : h(x) \neq f(x)\})$$

2.2 Empirical Risk Minimization

As mentioned earlier, in PAC learning a learner receives as input a training set $\mathcal{S}^m$, sampled from an unknown distribution $\mathcal{D}$ and labeled by some unknown target function f, and should output a predictor $h_{\mathcal{S}} : \mathcal{X} \to \mathcal{Y}$ (the subscript $\mathcal{S}$ emphasizes the fact that the output predictor depends on $\mathcal{S}^m$). The goal of the algorithm is to find $h_{\mathcal{S}}$ that minimizes the error with respect to the unknown $\mathcal{D}$ and f. Since the learner does not know what $\mathcal{D}$ and f are, the true error is not directly available to the learner. A useful notion of error that can be calculated by the learner is the training error – the error the classifier incurs over the training sample:

$$L_{\mathcal{S}}(h) := \frac{|i \in [m] : h(x_i) \neq y_i|}{m}$$

Since the training sample is the snapshot of the world that is available to the learner, it makes sense to search for a solution that works well on that data. This learning paradigm – coming up with a predictor h that minimizes $L_{\mathcal{S}}(h)$ – is called Empirical Risk Minimization or *ERM* for short.

2.3 Objective

Let $h_{\mathcal{S}}$ denote a result of applying $ERM_{\mathcal{H}}$ to $\mathcal{S}^m$, namely,

$$h_{\mathcal{S}} \in \underset{h \in \mathcal{H}}{argmin}\ L_{\mathcal{S}}(h)$$

Since $L_{\mathcal{D},f}(h_{\mathcal{S}})$ depends on the training set, $\mathcal{S}^m$, and that training set is picked by a random process, there is randomness in the choice of the predictor $h_{\mathcal{S}}$ and, consequently, in the risk $L_{\mathcal{D},f}(h_{\mathcal{S}})$. Formally, we say that it is a random variable. It is not realistic to expect that with full certainty $\mathcal{S}^m$ will suffice to direct the learner toward a good classifier (from the point of view of $\mathcal{D}$), as there is always some probability that the sampled training data happens to be very nonrepresentative of the underlying $\mathcal{D}$. Therefore it is addressed by the probability to sample a training set for which $L_{\mathcal{D},f}(h_{\mathcal{S}})$ is not too large. Usually, the probability of getting a nonrepresentative sample is denoted by δ, and is called $(1-\delta)$, the confidence parameter of the prediction.

On top of that, since there is no guarantee for perfect label prediction, another parameter is introduced for the quality of prediction, the accuracy parameter ϵ. The event $L_{\mathcal{D},f}(h_{\mathcal{S}}) > \epsilon$ is interpreted as a failure of the learner, while if $L_{\mathcal{D},f}(h_{\mathcal{S}}) \leq \epsilon$ the output of the algorithm is viewed as an approximately correct predictor. Therefore, the upper bounding the probability to sample m-tuple of instances that will lead to failure of the learner is of interest. Formally, let $\mathcal{S}^m|_x = (x_1, \cdots, x_m)$ be the instances of the training set. The desired upper bound is:

$$\mathcal{D}^m(\{\mathcal{S}|_x : L_{D,f}(h_{\mathcal{S}}) > \epsilon\}) \leq \delta \tag{1}$$

3 Compression Schemes

3.1 Motivation

This approach explores the PAC learnability using the paradigm of Data Compression. A first algorithm chooses a small subset of the sample which is called the Kernel (Compressor [1]). A second algorithm predicts future values of the function from the kernel, i.e. the algorithm

[1] This term also was not used in the original paper but we found it more revealing and easier intuitively to call this function this way.

acts as an hypothesis for the function to be learned. The second algorithm must be able to reconstruct the correct function values when given a point of the original sample thus, it's called Reconstructor (Decompressor [2].) The reason behind studying this framework as we will discuss in Sections 4.2 and 4.3 is deriving better sample complexities for PAC learnable concept classes than what [7] introduced for only finite concept classes and later [1] bettered it by introducing sample complexities even for infinite concept classes but, was limited to those which have finite VC dimension.

3.2 Formal model

Formally in this model, there is a function called Compressor κ such that $\kappa : \bigcup_{m=k}^{\infty} \mathcal{S}^m \to \mathcal{S}^k$ and simultaneously, there is another function called Decomporessor ρ such that $\rho : \mathcal{S}^k \times \mathcal{X} \to \mathcal{Y} = \{0, 1\}$. Now, the original objective defined in Equation 1 is turned to:

$$\mathcal{D}^m(\{\mathcal{S}|_x : L_{D,f}(\rho(\kappa(\mathcal{S}^m), x)) > \epsilon\}) \leq \delta \tag{2}$$

A data compression scheme of this form can be used as the basis of a learning algorithm. Given a labeled sample, $\mathcal{S}^m$, the algorithm makes the hypothesis that the concept is the set $\{x : \rho(\kappa(\mathcal{S}^m)), x) = 1\}$. Determining the compressor with k corresponds to computing the hypothesis, i.e. the compressor encodes the hypothesis. The computation of the value of the hypothesis is achieved with ρ using the compressor as an input. To fulfill the condition for polynomial PAC learnability the algorithms κ and ρ must be polynomial in the length of their input and the sample size m must be polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. What should be showed is that whenever there is a compression scheme with fixed kernel size [3] then m is always polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

3.3 Conditions

Here are the required conditions that should hold in a compression scheme:

1. $\forall m \geq k$, $\mathcal{S}^k$ is a subsequence of length k of $\mathcal{S}^m$
2. $\forall m, x_i \in \mathcal{S}^m|_x \implies \rho(\kappa(\mathcal{S}^m), x_i) = f(x_i)$

The second condition specifies that ρ reconstructs the labels of the sample points correctly. Usually both mappings are given by algorithms. It is also assumed that the decompressor ρ is Borel measurable. (This holds, for example, for functions on $\mathbb{R}^n$ built recursively from ordinary comparison and arithmetic operations. We also assume that the concepts in $\mathcal{H}$ are Borel measurable.)

[2] This term was not used in the original paper but we found it more revealing and easier intuitively to call this function this way.

[3] The kernel size of a concept class is defined as the minimum kernel size of all compression schemes.

4 Compression Schemes and PAC Learnability

4.1 Union bounds

Theorem 1. *For any compression scheme with kernel size k the error is larger than ϵ with probability (w.r.t. $\mathcal{D}^m$) less than $\binom{m}{k}(1-\epsilon)^{m-k}$ when given a sample of size $m \geq k$.*

Proof. Suppose we are learning some concept f. Given an ϵ and an m, we want to find a bound on the probability of choosing an m-sample which leads to a hypothesis with error greater than ϵ. In other words, we want to bound the error probability $\mathcal{D}^m(E)$ where

$$E = \{\mathcal{S}^m : \mathcal{D}(\{x : \rho(\kappa(\mathcal{S}^m), x) \neq f(x)\}) > \epsilon\}$$

Equivalently,

$$E = \{\mathcal{S}^m : \mathcal{D}(\{x : \rho(\kappa(\mathcal{S}^m), x) = f(x)\}) < 1-\epsilon\}$$

Let T be the collection of all k-element subsequences of the sequence $(1, 2, \cdots, m)$. For any $\bar{t} = (t_1, \cdots, t_k) \in T$, let

$$\begin{aligned} A_{\bar{t}} &= \{\mathcal{S}^m : \kappa(\mathcal{S}^m) = \mathcal{S}^k\} \\ E_{\bar{t}} &= \{\mathcal{S} \in A_{\bar{t}} : \mathcal{D}(\{x : \rho(\kappa(\mathcal{S}), x) = f(x)\}) < 1-\epsilon\} \\ U_{\bar{t}} &= \{\mathcal{S}^m : \mathcal{D}(\{x : \rho(\mathcal{S}^k, x) = f(x)\}) < 1-\epsilon\} \\ B_{\bar{t}} &= \{\mathcal{S}^m : \text{mark } \rho(\mathcal{S}^k, x_i) = f(x), \forall x_i \text{ s.t. } i \notin \bar{t})\} \end{aligned}$$

We have $E_{\bar{t}} = E \cap A_{\bar{t}}$ and since $\mathcal{S}^m = \bigcup_{\bar{t} \in T} A_{\bar{t}}, E = \bigcup_{\bar{t} \in T} E_{\bar{t}}$. From the definition of $A_{\bar{t}}$ we get

$$E_{\bar{t}} = \{\mathcal{S} \in A_{\bar{t}} : \mathcal{D}(\{x : \rho(\mathcal{S}^k, x) = f(x)\}) < 1-\epsilon\}$$

Thus $E_{\bar{t}} = U_{\bar{t}} \cap A_{\bar{t}}$. The second condition of Compression Schemes conditions 3.3 guarantees that $A_{\bar{t}} \subseteq B_{\bar{t}}$. Roughly, these sets serve us as follows: We split $\mathcal{S}^m$ into the $A_{\bar{t}}$ (which only overlap where m-samples have repeated points). We then look at the intersection of E with each of these $A_{\bar{t}}$. Extending these intersections to the sets $U_{\bar{t}} \cap B_{\bar{t}}$ eliminates explicit dependence of the sets on κ and gives us sets whose probabilities can be easily bounded. We have

$$\mathcal{D}^m(E_{\bar{t}}) \leq \mathcal{D}^m(U_{\bar{t}} \cap B_{\bar{t}})$$

It will now be convenient to rearrange the coordinates. Let $\pi_{\bar{t}}$ be any permutation of $1, 2, \cdots, m$ which sends i to ti, for $i = 1, \cdots, k$. Let $\phi_{\bar{t}} : \mathcal{S}^m \to \mathcal{S}^m$ send $(s_1, s_2, \cdots, s_m)$ to $(s_{\pi_{\bar{t}}(1)}, s_{\pi_{\bar{t}}(2)}, \cdots, s_{\pi_{\bar{t}}(m)})$ where $s_i = (x_i, y_i)$. We have

$$\mathcal{D}^m(U_{\bar{t}} \cap B_{\bar{t}}) = \mathcal{D}^m(\phi_{\bar{t}}(U_{\bar{t}}) \cap \phi_{\bar{t}}(B_{\bar{t}})) = \int_{\phi_{\bar{t}}(U_{\bar{t}})} I_{\phi_{\bar{t}}(B_{\bar{t}})} d\mathcal{D}^m$$

Note that

$$\phi_{\bar{t}}(U_{\bar{t}}) = \{\mathcal{S}^m : D(\{x : \rho(\mathcal{S}^k, x) = f(x)\}) < 1 - \epsilon\}$$

Thus there exists some set $V_{\bar{t}} \subset \mathcal{S}^k$ such that $\phi_{\bar{t}}(U_{\bar{t}}) = V_{\bar{t}} \times \mathcal{S}^{m-k}$. By Fubini's theorem we have

$$\int_{\phi_{\bar{t}}(U_{\bar{t}})} I_{\phi_{\bar{t}}(B_{\bar{t}})} d\mathcal{D}^m = \int_{V_{\bar{t}}} d\mathcal{D}^k \int_{\mathcal{S}^{m-k}} I_{\phi_{\bar{t}}(B_{\bar{t}})} d\mathcal{D}^{m-k} \tag{3}$$

We previously defined that

$$\phi_{\bar{t}}(B_{\bar{t}}) = \{\mathcal{S}^m : \rho(\mathcal{S}^k, x_i) = f(x_i) \ for \ i = k+1, \cdots, m\}$$

Let

$$W_{x_1, \cdots, x_k} = \{x \in \mathcal{X} : \rho(\mathcal{S}^k, x) = f(x)\}$$

Now

$$(s_1, \cdots, s_m) \times \mathcal{S}^{m-k} \cap \phi_{\bar{t}}(B_{\bar{t}}) = (s_1, \cdots, s_m) \times W^{m-k}_{x_1, \cdots, x_k}$$

Thus the inner integral in Equation 3 equals $\mathcal{D}^{m-k}(W^{m-k}_{x_1, \cdots, x_k})$. Since $(s_1, \cdots, s_k) \in V_{\bar{t}}$, we have $\mathcal{D}(W_{x_1, \cdots, x_k}) < 1 - \epsilon$. Thus the inner integral is bounded by $(1-\epsilon)^{m-k}$. This then bounds the entire integral, and we get

$$\mathcal{D}^m(E_{\bar{t}}) \leq (1-\epsilon)^{m-k}$$

Since the size of T is $\binom{m}{k}$, we have

$$\mathcal{D}^m(E) \leq \binom{m}{k}(1-\epsilon)^{m-k} \tag{4}$$

Which bounds our desired objective Equation 2 and the proof is completed. □

4.2 Sample complexity

In the following theorem we give explicit bounds for the sample size that guarantee PAC learnability. A similar bound $m \geq \max(\frac{8d}{\epsilon}\ln(\frac{8d}{\epsilon}), \frac{4}{\epsilon}\ln(\frac{2}{\delta}))$ was given in [1], where d denotes the Vapnik-Chervonenkis dimension of the class to be learned.

Theorem 2. *Any compression scheme with kernel-size $k \geq 1$ produces with probability at least $1 - \delta$ a hypothesis with error at most ϵ when given a sample of size*

$$m \geq \max(\frac{4k}{\epsilon}\ln(\frac{4k}{\epsilon}) + 2k, \frac{2}{\epsilon}\ln(\frac{1}{\delta}))$$

This holds for arbitrary ϵ and δ.

Proof. Follows from the bound of the Equation 4. Applying the previous theorem it suffices to show that if m fulfills the bound then $\binom{m}{k}(1-\epsilon)^{m-k} \leq \delta$.

So

$$\delta \geq \binom{m}{k}(1-\epsilon)^{m-k} \Longrightarrow (1-\epsilon)^{-(m-k)} \geq \binom{m}{k}\frac{1}{\delta} \Longrightarrow$$

$$-(m-k)ln(1-\epsilon) \geq ln(\frac{1}{\delta}) + ln(\binom{m}{k}) \Longrightarrow m \geq \frac{ln(\frac{1}{\delta}) + ln(\binom{m}{k})}{-ln(1-\epsilon)} + k \tag{5}$$

Also, We know that

$$\binom{m}{k} \leq \frac{m^k}{k!} \leq m^k$$

And

$$1-\epsilon \leq e^{-\epsilon} \Longrightarrow -ln(1-\epsilon) \geq \epsilon$$

Thus Equation 5 always holds if

$$m \geq \frac{ln(\frac{1}{\delta}) + ln(m^k)}{\epsilon} + k = \frac{1}{\epsilon}(\ln(\frac{1}{\delta})) + k(\frac{1}{\epsilon}ln(m)+1) \tag{6}$$

There are two summands in Equation 6. The inequality certainly holds if each summand is at most $\frac{m}{2}$. For the first summand this easily leads to the first bound in the maximum expression of the theorem. Similarly the second summand will lead to the second bound of the theorem. If

$$\frac{m}{2} \geq k(\frac{1}{\epsilon}ln(m)+1)$$

Replacing m by the second bound in the above inequality leads to

$$\frac{2k}{\epsilon}ln(\frac{4k}{\epsilon}) + k \geq \frac{k}{\epsilon}(ln(\frac{4k}{\epsilon}) + ln(ln(\frac{4k}{\epsilon}) + \frac{\epsilon}{2})) + k$$

which simplifies to $\frac{4k}{\epsilon} \geq ln(\frac{4k}{\epsilon}) + \frac{1}{2}$ and can be easily verified. □

4.3 Compression Scheme vs VC dimension

Now it's a good time to compare the sample complexity obtained in Equation 2 to the $m \geq \max(\frac{8d}{\epsilon}\ln(\frac{8d}{\epsilon}), \frac{4}{\epsilon}\ln(\frac{2}{\delta}))$ that was introduced previously by [1]. In the example of learning n-dimensional orthogonal rectangles the VC dimension is $2n$ and The kernel size of the straight forward compression scheme is also $2n$. Thus the bounds stated in the previous theorem are better roughly by a factor of two. More importantly, the VC dimension and the kernel size are not always equal. In the case of arbitrary halfplanes the VC dimension is three but there exists an algorithm with kernel size two [2]!

4.4 Improving the Sample Complexity

The bound obtained in Equation 4 can also be bettered. [3] changed the perspective slightly by imposing a restricter condition that is instead of having a fixed k let's say that the compression scheme could at most a kernel size of k then the following bound naturally arises

$$\mathcal{D}^m(E) \leq \sum_{i=0}^{k} \binom{m}{i} (1-\epsilon)^{m-i}$$

then the following lemma gives the new weaker bounds.

Lemma 1. *For* $0 \leq \epsilon, \delta \leq 1$*, if*

$$m \geq \frac{1}{(1-\beta)}(\frac{1}{\epsilon}\ln(\frac{1}{\delta}) + k + \frac{k}{\epsilon}ln(\frac{1}{\beta\epsilon})) \tag{7}$$

for any $0 < \beta < 1$ *then* $\mathcal{D}^m(E) \leq \sum_{i=0}^{k} \binom{m}{i}(1-\epsilon)^{m-i} \leq \delta$

Proof. Let

$$m \geq \frac{1}{(1-\beta)}(\frac{1}{\epsilon}\ln(\frac{1}{\delta}) + k + \frac{k}{\epsilon}ln(\frac{1}{\beta\epsilon}))$$

for $0 < \beta < 1$ which is equivalent to

$$\frac{1}{\epsilon}ln(\frac{1}{\delta}) + k + \frac{k}{\epsilon}(1 + ln(\frac{k}{\beta\epsilon}) - 1 + \frac{\beta\epsilon}{k}m - ln(k)) \leq m \tag{8}$$

By using the fact from [6] that

$$-ln(\alpha) - 1 + \alpha m \leq \ln(m)\ \forall \alpha > 0$$

For $\alpha = \frac{\beta\epsilon}{k}$ we get

$$\ln(\frac{k}{\beta\epsilon}) - 1 + \frac{\beta\epsilon}{k}m \geq ln(m)$$

By substituting $\ln(m)$ into the left hand side of Equation 8 we get

$$\frac{1}{\epsilon}\ln(\frac{1}{\delta}) + k + \frac{k}{\epsilon}(1 + \ln(m) - \ln(k)) \leq m \implies \ln\frac{1}{\delta} + k(1 + \ln m - \ln k) \leq \epsilon(m-k)$$

$\implies (\frac{em}{k})^k \leq e^{\epsilon(m-k)}\delta$

Also we know that if m, d are two positive integers such that $d \leq m-2$. Then,

$$\sum_{k=0}^{d} \binom{m}{k} \leq (\frac{em}{d})^d$$

So we have

$$\sum_{i=0}^{k} \binom{m}{i}(1-\epsilon)^{m-i} \leq (\frac{em}{k})^k e^{-\epsilon(m-k)} \leq \delta \tag{9}$$

□

The above lemma leads to sample size bounds that grow linearly in the size of the compression scheme k. The original of [4] obtained in Equation 6 had $k \ln k$ dependence.

5 Uniform deviation bounds

Finally, in the following theorem we show that it is possible to derive uniform deviation bounds based on the compression schemes.

Theorem 3. *Consider a sample compression algorithm $\mathcal{A}$ of size k and a bounded loss function $\ell \in [0, c]$ and a dataset $\mathcal{S}$ of size m* [4] *then:*

$$\ell(\mathcal{A}(\mathcal{S}); \mathcal{D}) \leq \frac{m}{m-k} \ell(\mathcal{A}(\mathcal{S}); \mathcal{S}) + c\sqrt{\frac{\log \frac{1}{\delta} + k \log \frac{em}{k}}{2m}}$$

Proof. Let $I \subseteq \{1, \cdots, m\}$, with $|I| \leq k$.

Let $h_I = \rho(\mathcal{S}_I)$ where $\mathcal{S}_I = \{(x_i, y_i)\}_{i \in I}$.

Note: h_I is independent of $\mathcal{S}_{\bar{I}}$, where $\bar{I} = \{1, \cdots, m\} \setminus I$.

By Hoeffding,

$$\ell(h_I; D) \leq \ell(h_I; \mathcal{S}_{\bar{I}}) + c\sqrt{\frac{\log \frac{1}{\delta}}{2m}} \text{ with probability} \geq 1 - \delta \tag{10}$$

The number of candidate output hypotheses of $\mathcal{A}$ is

$$\sum_{i=0}^{k} \binom{m}{i} \leq (\frac{em}{k})^k$$

Using union bound, Equation 10 holds for all $|I| \leq k$ uniformly with probability $\geq 1 - \delta(\frac{em}{k})^k = 1 - \delta'$.

Therefore, with probability $\geq 1 - \delta', \forall |I| \leq k$,

$$\ell(h_I; D) \leq \ell(h_I; \mathcal{S}_{\bar{I}}) + c\sqrt{\frac{\log \frac{1}{\delta'} + k \log \frac{em}{k}}{2m}}$$

Note that $(m-k)\ell(h; \mathcal{S}_{\bar{I}}) \leq m\ell(h; \mathcal{S})$ □

6 Conclusion

In this project, we first started from basics and introduced the framework of PAC learning. To this end, we introduced the formal components of this setting in Section 2 and defined the objectives that are of interest in it. Then, we introduced the framework of compression schemes, necessary conditions for its applicability in PAC learning. After that, we first derived the union bounds using compression schemes in Section 4.1 and based on that, obtained the new sample

[4] For simplicity we drop the the explicit dependency of the dataset to its size in its notation meaning that here, $\mathcal{S} \equiv \mathcal{S}^m$.

complexity 2 and secondly, in Section 4.3 we discussed whether the new bounds are really better than the previous bounds. Also, we showed that 2 can also be improved further and get even smaller bounds in Section 4.4 by Equation 7. And lastly, in Section 5 we derived uniform deviation bounds on the compression schemes.

The main takeaways could be summarized as how this body of work contains a string of connected topics and each piece of work added an improvement on top of a previous established result. Then, understanding what compression schemes were all about and, how we could still do much better instead of resorting to saying that having a finite VC dimension is a necessary and sufficient condition of PAC learnability! Yet on this front, we faced some questionable points like:

- *How much valuable are the new bounds if they only are better up to a constant factor? 4.3*
- *What are the necessary conditions for existence of a compression scheme? 3.2 Or in other words, How can someone design κ and ρ while meeting the required conditions instead of viewing these functions as givens?*
- *How was the accuracy parameter (ϵ) used in Section 4.1 to derive unions bounds and in the process it was used to bound a probability while it can be larger than 1?*

And, it was very useful to see how union bounds can be derived using set rules in Section 4.1 and, seeing that Hoeffding inequality also holds for compression schemes to derive uniform deviation bounds.

References

[1] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[2] A. Blumer and N. Littlestone. Learning faster than promised by the vapnik-chervonenkis dimension. *Discrete Applied Mathematics*, 24(1):47–53, 1989.

[3] S. Floyd and M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine learning*, 21:269–304, 1995.

[4] N. Littlestone and M. K. Warmuth. Relating data compression and learnability. 2003.

[5] S. Moran and A. Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM (JACM)*, 63(3):1–10, 2016.

[6] J. Shawe-Taylor, M. Anthony, and R. L. Biggs. Bounding sample size with the vapnik-chervonenkis dimension. Technical Report CSD-TR-618, University of London, Royal Halloway and New Bedford College, 1989.

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