List Sample Compression and Uniform Convergence

Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Department of Computer Science, Purdue University

Shay Moran SMORAN@TECHNION.AC.IL
Departments of Mathematics, Computer Science, and Data and Decision Sciences, Technion and Google Research

Tom Waknine TOM.WAKNINE@CAMPUS.TECHNION.AC.IL
Department of Mathematics, Technion

Editors: Shipra Agrawal and Aaron Roth

Abstract

List learning is a variant of supervised classification where the learner outputs multiple plausible labels for each instance rather than just one. We investigate classical principles related to generalization within the context of list learning. Our primary goal is to determine whether classical principles in the PAC setting retain their applicability in the domain of list PAC learning. We focus on uniform convergence (which is the basis of Empirical Risk Minimization) and on sample compression (which is a powerful manifestation of Occam's Razor). In classical PAC learning, both uniform convergence and sample compression satisfy a form of 'completeness': whenever a class is learnable, it can also be learned by a learning rule that adheres to these principles. We ask whether the same completeness holds true in the list learning setting.

We show that uniform convergence remains equivalent to learnability in the list PAC learning setting. In contrast, our findings reveal surprising results regarding sample compression: we prove that when the label space is $\mathcal{Y} = \{0, 1, 2\}$, then there are 2-list-learnable classes that cannot be compressed. This refutes the list version of the sample compression conjecture by Littlestone and Warmuth (1986). We prove an even stronger impossibility result, showing that there are 2-list-learnable classes that cannot be compressed even when the reconstructed function can work with lists of arbitrarily large size. We prove a similar result for (1-list) PAC learnable classes when the label space is unbounded. This generalizes a recent result by Pabbaraju (2023).

In our impossibility results on sample compression, we employ direct-sum arguments which might be of independent interest. In fact, these arguments raise natural open questions that we leave for future research. Our findings regarding uniform convergence rely on a coding theoretic perspective.

1. Introduction

List learning is a natural generalization of supervised classification, in which, instead of predicting the correct label, the learner outputs a small list of labels, one of which should be the correct one. This approach can be viewed as giving the learner more than one guess at the correct label.

There are many settings in which one may prefer the list learning approach to the classical one. For example, recommendation systems often suggest a short list of products to users, with the hope that the customer will be interested in one of them (see Figure 1). Another example is the top-k loss function in which the model gets k guesses for each sample. This loss function is often used in ML competitions and can be seen as a variant of list learning. Additionally, list learning addresses label ambiguity; for example, in computer vision recognition problems, it is often impossible to determine if a certain image is of a pond or a river. As a result, training a model for such problems by penalizing it for every mistake can be too restrictive. However, using a top-k approach seems like a reasonable alternative. This approach has been studied in recent works such as Lapin, Hein, and Schiele (2015) and Yan, Luo, Liu, Li, and Zheng (2018), which demonstrate its usefulness in certain problems.

List learning has also found applications in theoretical machine learning. For example in Brukhim, Carmon, Dinur, Moran, and Yehudayoff (2022) it was an essential part of establishing the equivalence

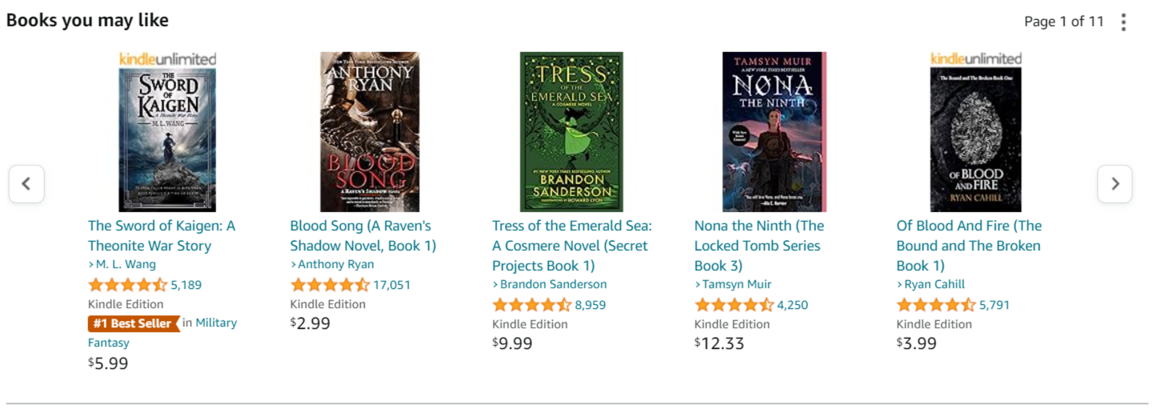

Figure 1: Amazon recommendation system gives their users a short list of books based on their past reading, aiming that one of those books will capture their interest.

between finite Daniely-Shwartz (DS) dimension and multiclass learnability. Consequently, list learning has been studied in several recent works in learning theory. For example, Charikar and Pabbaraju (2022) characterized list PAC learnability by using a list variant of the DS dimension, and Moran, Sharon, Tsubari, and Yosebashvili (2023) characterized list online learnability using a list variant of the Littlestone dimension. Another recent application of list learning is in the realm of multiclass boosting; Brukhim, Hanneke, and Moran (2023) employed it to devise the first boosting algorithm whose sample complexity is independent of the label space's size.

A natural question that has not yet been systematically addressed is the identification of fundamental principles in list PAC learning. In the binary case, PAC learning is guided by fundamental algorithmic principles, notably Empirical Risk Minimization, and Occam's Razor principles such as compression bounds. In this work, we ask which of these foundational principles remains applicable in the domain of list learning.

1.1. Our Contribution

In this section we summarize our main results. It is based on natural adaptations of basic learning theoretic definitions to the list setting. These definitions are fully stated in Section 2.

1.1.1. Sample Compression

Occam's Razor. Occam's razor is a philosophical principle applied broadly across disciplines, including machine learning. It suggests that among competing hypotheses, the simplest one should be selected. In machine learning, this principle is often quantified in terms of the number of bits required to encode an hypothesis, thereby serving as a guideline for preferring simpler models. This concept forms the basis of a general approach in machine learning, where simplicity is directly linked to the efficiency and effectiveness of learning algorithms.

A more refined manifestation of Occam's razor in machine learning is evident in sample compression schemes. These schemes go beyond merely considering the bit-size of an hypothesis. They involve an additional use of a small, representative subset of input examples to encode hypotheses. This approach is a more nuanced application of the principle, allowing for both the simplification of data and the preservation of its essential characteristics. In fact, while the classical interpretation of Occam's razor, based on bit-encoding, is sound — implying generalization — it lacks completeness. This means that there exist learnable classes

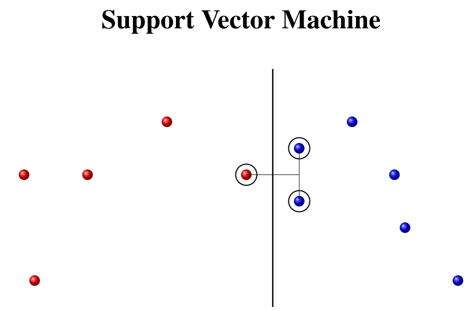

Figure 2: Support Vector Machine (SVM) as sample compression: The SVM algorithm identifies a maximum-margin separating hyperplane, which is defined by a subsample of $d+1$ support points. This exemplifies a concept class which cannot be learned with hypotheses specified by finite bit-encoding, yet for which there is an algorithm which learns effectively using hypotheses specified using a bounded number of input-examples.

whose learnability cannot be demonstrated solely through bit-encoding (see Figure 2)[1]. In contrast, sample compression schemes offer a more comprehensive manifestation of Occam's razor: every PAC learnable class can be effectively learned by a sample compression algorithm (David, Moran, and Yehudayoff, 2016).

List Sample Compression Schemes. Sample compression schemes were initially developed by Littlestone and Warmuth (1986) for the purpose of proving generalization bounds; however, they can also be interpreted as a standalone mathematical model for *data simplification*. Sample compression resembles a scientist who collects extensive experimental data but then selects only a crucial, representative subset from it. From this subset, a concise hypothesis is formulated that effectively explains the entire dataset. This analogy highlights the essence of sample compression: distilling a complex dataset into a simpler form, while maintaining the capacity to accurately explain the patterns and phenomena of the entire original dataset.

More formally, a compression scheme consists of a pair of functions: a *compressor* and a *reconstructor*. The compressor gets an input sample of labeled examples S and uses it to construct a small subsequence of labeled examples S' which she sends to the reconstructor. The reconstructor uses S' to generate an concept h. The goal is that h will be consistent with the original sample S, even on examples that did not appear in S' (see Figure 3)[1].

We extend the notion of sample compression to the list setting naturally by changing the concept h outputted by the reconstructor to be a k-list map. Now the goal is that h will be consistent with S in the sense that for any example x, its label will be one of the k elements of the list $h(x)$. In more detail, a concept class $\mathcal{C}$ is k-list compressible if there exists a k-list sample compression scheme with a finite size such that whenever the input sample S is realizable[2] by $\mathcal{C}$ then the reconstructed k-list concept h satisfies $y_i \in h(x_i)$ for all $(x_i, y_i) \in S$. We refer to Section 2.2 for more details.

In their original paper Littlestone and Warmuth (1986) showed that sample compression schemes are PAC learners, and conjectured that every PAC learnable class admits a sample compression scheme. Moran and Yehudayoff (2016) confirmed this conjecture in the binary setting, and David et al. (2016) further showed that any PAC learnable class over a finite label space admits a sample compression scheme. It is therefore natural to ask whether the analogue of Littlestone and Warmuth (1986) conjecture holds in the list setting:

Does every k-list learnable class has a finite k-list sample compression scheme?

1. Figures 2 and 3 from Alon, Hanneke, Holzman, and Moran (2021), used with permission.
2. I.e. there exists $c \in \mathcal{C}$ such that $c(x_i) = y_i$ for all $(x_i, y_i) \in S$.

A pictorial definition of a sample compression scheme

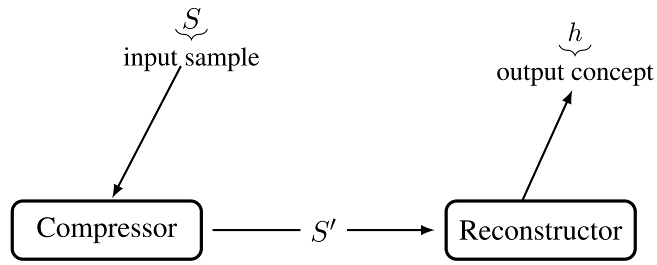

Figure 3: An illustration of a sample compression scheme: S' is a subsample of S; the output concept h should be consistent with the entire input sample S. In the context of k-list sample compression, the concept h assigns a list of k possible labels to each data point x. The objective is refined such that for every input example (x_i, y_i), the actual label y_i is included within the list provided by $h(x_i)$.

Our first result provides a negative answer in the simplest list PAC learning setting: 2-list learning a 3-label space:

Theorem 1 (List-Learnability vs. List-Compressibility) *There exists a concept class $\mathcal{C}$ over the label space $\mathcal{Y} = \{0, 1, 2\}$ such that:*

- *$\mathcal{C}$ is 2-list PAC learnable.*
- *$\mathcal{C}$ has no finite 2-list sample compression scheme.*

Reconstruction with Larger Lists. The class $\mathcal{C}$ in Theorem 1 is trivially 3-list compressible: simply take the reconstructed concept h be such that $h(x) = \{0, 1, 2\}$ for every x. In a related and recent result, Pabbaraju (2023) considers the setting of multiclass PAC learning over an infinite label space and establishes the existence of a concept class $\mathcal{C}$ that is (1-list) PAC learnable but not (1-list) compressible. Also in his construction, the class $\mathcal{C}$ is trivially 2-list compressible. These examples raise the following question:

Does every k-list PAC learnable class $\mathcal{C}$ admit a $(k+1)$-list sample compression scheme?

We not only answer this question in the negative but also prove a significantly stronger result. We demonstrate the existence of learnable classes that are not k-compressible for any arbitrarily large k:

Theorem 2 *[2-Learnability vs k-Compressibility] For any $k > 0$ there exists a concept class $\mathcal{C}_k$ over a finite label space $\mathcal{Y}_k$ that satisfies the following:*

1. *$\mathcal{C}_k$ is 2-list PAC learnable.*
2. *$\mathcal{C}_k$ has no finite k-list sample compression scheme.*

Theorem 3 *[1-Learnability vs k-Compressibility] For any $k > 0$ there exists a concept class $\mathcal{C}_k$ satisfying the following:*

1. *$\mathcal{C}_k$ is PAC learnable. (I.e. 1-list PAC learnable)*
2. *$\mathcal{C}_k$ has no finite k-list sample compression scheme.*

Note that by David, Moran, and Yehudayoff (2016) the label space of the class in Theorem 3 is inevitably infinite and that the case $k = 1$ gives the aforementioned result of Pabbaraju (2023). Additionally, a close examination of the proofs for these theorems reveals that in both scenarios, the respective classes lack a compression scheme of size $f(n)$ for any function $f(n) = o(\log n)$.

The proof of Theorem 1 is based on a construction from Alon, Hanneke, Holzman, and Moran (2021) of a learnable partial concept class which is not compressible. This is similar to how Pabbaraju (2023) derives a non compressible PAC learnable class over an infinite label space; however, the argument is somewhat more involved because Theorem 1 provides a consturction over a finite label space, whereas in Pabbaraju (2023) the label space is inevitably infinite. The proofs of Theorem 2 and Theorem 3 are by induction on k, where the base case reduces to the partial concept class construction from Alon et al. (2021). The induction step, however, requires new ideas and will use direct sum arguments that we develop in appendix A.2 and overview in Section 1.1.3. We give a more detailed review of the proof method in Section 3.1.

1.1.2. UNIFORM CONVERGENCE

Uniform convergence and empirical risk minimization (ERM) are arguably the most extensively studied algorithmic principles for generalization in machine learning. For example, the Fundamental Theorem of PAC Learning for binary classification states the equivalence between PAC learnability, uniform convergence, and the ERM principle (Shalev-Shwartz and Ben-David, 2014). In fact, the equivalence between uniform convergence and PAC learnability holds whenever the label space is finite (Shalev-Shwartz, Shamir, Srebro, and Sridharan, 2010). Moreover, ERM is closely related to other statistical principles such as maximum likelihood estimation.

Informally, uniform convergence refers to the phenomenon where, given a sufficiently large sample from a population, the empirical losses of all concepts in a class closely approximate their true losses over the entire population. This phenomenon forms the basis of the Empirical Risk Minimization principle. It posits that selecting a concept in the class that minimizes empirical loss is a sound strategy, as such a concept is also likely to approximately minimize the population loss.

In the context of list learning, we consider classes $\mathcal{C}$ of k-list concepts. So, each $c \in \mathcal{C}$ assigns a list of k labels to each point $x \in \mathcal{X}$. For a target distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the population loss of c is $\mathtt{L}_{\mathcal{D}}(c) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[1_{y\notin c(x)}]$ and for a sample $S = \{(x_i, y_i)\}_{i=1}^n$, the empirical loss of c is $\mathtt{L}_S(c) = \frac{1}{n}\sum_{i=1}^n 1[y_i \notin c(x_i)]$. We investigate when k-list concept classes exhibit uniform convergence and ask whether a parallel to the Fundamental Theorem of PAC Learning exists in this setting.

Does the equivalence between PAC learning and uniform convergence extend to k-list concept classes?

We confirm this is the case, providing an affirmative response to these questions, though via a novel analysis deviating significantly from the traditional approach (as outlined below). Please see Section 2.1 for the definitions of PAC learnability and uniform convergence, adapted to the list learning setting.

Theorem 4 (List Learnability vs. Uniform Convergence) *Let $\mathcal{C} \subset \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ be a k-list concept class over a finite label space $|\mathcal{Y}| < \infty$. Then, the following properties are equivalent:*

- *$\mathcal{C}$ is k-list PAC learnable.*
- *$\mathcal{C}$ is k-list agnostically PAC learnable.*
- *$\mathcal{C}$ satisfies the uniform convergence property.*

In Theorem 37 in Section B we also provide quantitative bounds on the uniform convergence rate. These bounds follow from analyzing the $\mathtt{DS}_k$ dimension (which controls the learning rate) and the graph dimension (which controls the uniform convergence rate); see Section 2 for further details. In particular, our result implies $\mathtt{DS}_k(\mathcal{C}) = \tilde{\Omega}\left(\frac{\mathtt{G}_k(\mathcal{C})}{k\cdot\log(|\mathcal{Y}|\cdot\mathtt{G}_k)}\right)$[3]. This implies the following upper bound on the uniform convergence rate:

$$\varepsilon(n|\mathcal{C}) = \tilde{O}\left(\sqrt{\frac{k \cdot \mathtt{DS}_k(\mathcal{C}) \cdot \log(|\mathcal{Y}| \cdot \mathtt{DS}_k(\mathcal{C}))}{n}}\right).$$

This finding extends the ERM principle in the realm of list learning: for any class of k-list concepts that is k-list learnable, an effective learning strategy is to choose a concept from the class that minimizes the empirical loss.

The assumption that the label space $\mathcal{Y}$ is finite is necessary. Indeed, Daniely, Sabato, Ben-David, and Shalev-Shwartz (2015) demonstrate that already in the case of $k = 1$, there are PAC learnable classes that do not satisfy uniform convergence.

The proof of Theorem 4 deviates from the classical approaches to deriving uniform convergence. Typically, these bounds are obtained using a ghost sample argument combined with a growth function bound for the concept class $\mathcal{C}$. However, in the k-list learning context, some list-learnable classes $\mathcal{C}$ exhibit growth functions that are excessively large to apply this method. To overcome this, we directly analyze the VC dimension of the loss functions. Utilizing a probabilistic argument, we demonstrate that a high VC dimension of the loss function directly implies a significantly large k-DS dimension for the class $\mathcal{C}$. Consequently, if the class $\mathcal{C}$ does not satisfy uniform convergence, it is not PAC learnable.

1.1.3. DIRECT SUM

In computer science, the term 'direct sum' refers to fundamental questions about the scaling of computational or information complexity with respect to multiple task instances. Consider an algorithmic task T and a computational resource C. For instance, T might be the task of computing a polynomial, with C representing the number of arithmetic operations required, or T could be a learning task with its sample complexity as C. The direct sum inquiry focuses on the cost of solving k separate instances of T, particularly how this aggregate cost compares to the resources needed for a single instance. Typically, the cost for multiple instances is at most k times the cost of one, since each can be handled independently.

However, there are intriguing scenarios where the total cost for k instances is less than this linear relationship. These cases suggest more efficient methods for simultaneously handling multiple instances of a task than addressing them one by one. As an example, consider an $n \times n$ matrix A and the objective of calculating its product with an input column vector x, where the computational resource C is the number of arithmetic operations. For a single vector x, it is easy to see that $\Theta(n^2)$ operations are necessary and sufficient. However, if instead of one input vector x, there are n input vectors $x_1, \dots, x_n$ then one can do better than $n \times \Theta(n^2) = \Theta(n^3)$. Indeed, by arranging these n vectors as columns in an $n \times n$ matrix B, computing the product $A \cdot B$ is equivalent to solving the n products $A \cdot x_i$. This task can be accomplished using roughly $n^\omega \leq n^{2.37}$ arithmetic operations with fast matrix multiplication algorithms. Direct sum questions are well-studied in information theory and complexity theory. For more background we refer the reader to the thesis by Pankratov (2012) or the books by Wigderson (2019) and Rao and Yehudayoff (2020).

Direct Sum in Learning Theory. Natural direct sum questions can also be posed in learning theory. To formalize these, we use the notion of cartesian product of concept classes: consider two concept classes, $\mathcal{C}_1$ and $\mathcal{C}_2$, defined over domains $\mathcal{X}_1$ and $\mathcal{X}_2$, and label spaces $\mathcal{Y}_1$ and $\mathcal{Y}_2$ respectively. Their product, $\mathcal{C}_1 \otimes \mathcal{C}_2$, has domain $\mathcal{X}_1 \otimes \mathcal{X}_2$ and label space $\mathcal{Y}_1 \otimes \mathcal{Y}_2$. Each concept c in $\mathcal{C}_1 \otimes \mathcal{C}_2$ is parameterized by a pair of concepts $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$, and is defined as $c((x_1, x_2)) = (c_1(x_1), c_2(x_2))$. Thus, learning c effectively means learning both c_1 and c_2 simultaneously.

3. Here $\mathtt{DS}_k$ and $\mathtt{G}_k$ denote the Daniely-Shwartz and Graph Dimensions respectively. see definitions 14 and 12.

In our proofs of Theorems 2 and 3, we study the sample compression complexity of such product classes. While direct sum analysis serves primarily as a technical instrument in our research, it also leads to basic questions that we propose for future research. For instance, consider the following question:

Open Question 1 (Direct Sum: PAC Learning Curves) *Let $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and let $\varepsilon(n|\mathcal{C})$ denote the realizable PAC learning curve of $\mathcal{C}$ (see Definition 6). For $r \in \mathbb{N}$ let $\mathcal{C}^r = \Pi_{i=1}^r \mathcal{C}$ be the r-fold Cartesian product of $\mathcal{C}$. By a union bound, learning each component independently gives*

$$\varepsilon(n|\mathcal{C}^r) \leq r \cdot \varepsilon(n|\mathcal{C}).$$

Can the upper bound be asymptotically improved for some classes $\mathcal{C}$?

Further discussion and open questions related to the direct sums of learning problems are elaborated in appendix C.

Organization. The remainder of this paper is organized as follows: Section 2 presents the fundamental definitions of PAC learnability and sample compression, adapted for the list learning setting. Subsequently, Section 3 provides an overview of the techniques and key ideas employed in our proofs. Then, in Appendix A we give proofs for results on sample compression, and in Appendix B the uniform convergence results are proven. Finally in Appendix C we look deeper into the direct sum questions in learning theory and propose some open questions and directions for future research.

2. Basic Definitions

2.1. Generalization

We use standard notation from learning theory, see e.g. Shalev-Shwartz and Ben-David (2014). Let $\mathcal{X}$ denote the domain and $\mathcal{Y}$ denote the label space. A k-list function (or k-list concept) is a function $c : \mathcal{X} \to \binom{\mathcal{Y}}{k}$, where $\binom{\mathcal{Y}}{k}$ denotes the collection of all subsets of $\mathcal{Y}$ of size k. A k-list concept class $\mathcal{C} \subseteq \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ is a set of k-list functions. Note that by identifying sets of size one with their single elements, 1-list concept classes correspond to standard concept classes.

A k-list learning rule is a map $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \binom{\mathcal{Y}}{k}^{\mathcal{X}}$, i.e. it gets a finite sequence of labeled examples as input and outputs a k-list function. A learning problem $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$. The population loss of a k-list function c with respect to $\mathcal{D}$ is defined by $\mathrm{L}_{\mathcal{D}}(c) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[1_{y \notin c(x)}]$.

We quantify the learning rate of a given learning rule on a given learning problem using learning curves:

Definition 5 (Learning Curve) *The learning curve of a learner $\mathcal{A}$ with respect to a learning problem $\mathcal{D}$ is the sequence $\{\varepsilon_n(\mathcal{D}|\mathcal{A})\}_{n=1}^{\infty}$, where*

$$\varepsilon_n(\mathcal{D}|\mathcal{A}) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}\Big[\mathrm{L}_{\mathcal{D}}\big(\mathcal{A}(S)\big)\Big].$$

In words, $\varepsilon_n(\mathcal{D}|\mathcal{A})$ is the expected error of the learner $\mathcal{A}$ on samples of size n drawn from the distribution $\mathcal{D}$.

For a sequence S of labeled examples, the empirical loss of a k-list function c with respect to S is $\mathrm{L}_S(c) = \frac{1}{|S|}\sum_{(x,y)\in S} 1_{y\notin c(x)}$. A sequence $S \in (\mathcal{X} \times \mathcal{Y})^n$ is realizable by a k-list function c if $y \in c(x)$ for every $(x, y) \in S$. It is realizable by a concept class $\mathcal{C}$ if it is realizable by some concept $c \in \mathcal{C}$. A learning problem $\mathcal{D}$ is realizable by a concept class $\mathcal{C}$ if for any $n \in \mathbb{N}$, a random sample $S \sim \mathcal{D}^n$ is realizable by $\mathcal{C}$ with probability 1.

Definition 6 (List PAC Learnability) *We say that a concept class $\mathcal{C}$ is agnostically k-list learnable if there exists a k-list learning rule $\mathcal{A}$ and a sequence $\varepsilon_n \xrightarrow{n\to\infty} 0$ such that for every learning problem $\mathcal{D}$, $(\forall n) : \varepsilon_n(\mathcal{D}|A) \leq \inf_{c\in\mathcal{C}} \mathtt{L}_{\mathcal{D}}(c) + \varepsilon_n$. If the latter only holds for $\mathcal{C}$-realizable distributions then we say that $\mathcal{C}$ is k-list learnable in the realizable case. The k-list realizable PAC learning curve of a concept class $\mathcal{C}$ is defined as follows:*

$$\varepsilon(n|\mathcal{C}) = \inf_{\mathcal{A}} \sup_{\mathcal{D}} \varepsilon_n(\mathcal{D}|A),$$

where the infimum is taken over all k-list learning rules $\mathcal{A}$ and the supremum over all distributions $\mathcal{D}$ that are realizable by $\mathcal{C}$.

Observe that a class $\mathcal{C}$ is k-list PAC learnable in the realizable case if and only if its PAC learning curve approaches zero as $n \to \infty$.

Remark 7 *In the literature, PAC learnability is sometimes defined with a more stringent requirement: that the error is small with high probability, as opposed to merely the expected error in our definition. These two formulations are equivalent, as follows by a standard confidence amplification technique. We chose the above definition as it is simpler in that it involves fewer parameters (it omits the confidence parameter).*

In appendix A.1 we prove an equivalence between agnostic and realizable case learnability:

Theorem 8 *Let $\mathcal{C}$ be a concept class. Then $\mathcal{C}$ is k-list learnable in the realizable setting if and only if it is k-list learnable in the agonstic setting.*

Hence, we sometimes refer to a concept class as 'learnable' without distinguishing between the agnostic and realizable cases. The proof of Theorem 8 is a straight-forward adaptation of the proof of the parallel result for $k = 1$ David et al. (2016).

Our last definition in this section is of uniform convergence.

Definition 9 (Uniform Convergence) *We say that a k-list concept class $\mathcal{C}$ satisfies uniform convergence if there exists a vanishing sequence $\varepsilon_n \xrightarrow{n\to\infty} 0$ such that for all distributions $\mathcal{D}$,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}[\sup_{h\in\mathcal{C}}|\mathtt{L}_{\mathcal{D}}(h) - \mathtt{L}_S(h)|] \leq \varepsilon_n.$$

The uniform convergence rate of $\mathcal{C}$ is defined by $\varepsilon_{\mathtt{UC}}(n|\mathcal{C}) = \sup_{\mathcal{D}} \mathbb{E}_{S\sim\mathcal{D}^n}[\sup_{h\in\mathcal{C}}|\mathtt{L}_{\mathcal{D}}(h) - \mathtt{L}_S(h)|]$, where the supremum is over all distributions $\mathcal{D}$.

Note that $\mathcal{C}$ satisfies uniform convergence if and only if its uniform convergence rate converges to zero as $n \to \infty$. It is also worth mentioning that in the binary setting (i.e. when $\mathcal{Y} = \{0, 1\}$), PAC learnability is equivalent to uniform convergence, and that this equivalence justifies the Empirical Risk Minimization principle of outputting an concept in $\mathcal{C}$ which minimizes the sample error.

2.2. List Sample Compression

Definition 10 (List Compressibility) *A concept class $\mathcal{C}$ is agnostically k-list compressible if there exist $d \in \mathbb{N}$ and a reconstruction function $\rho : (\mathcal{X} \times \mathcal{Y})^d \to \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ such that the following holds. For every sample $S \in (\mathcal{X} \times \mathcal{Y})^\star$ there exists $S' = \big((x_1, y_1), \ldots (x_d, y_d)\big)$, where $(x_i, y_i) \in S$ for all $i \leq d$ such that $\mathtt{L}_S(\rho(S')) \leq \inf_{c\in\mathcal{C}} \mathtt{L}_S(c)$. If the latter only holds for $\mathcal{C}$-realizable samples then we say that $\mathcal{C}$ is k-list compressible in the realizable case.*

In some places, the map that takes S to S' is explicitly defined as the compression map and is denoted by κ, and the pair (ρ, κ) is called the sample compression scheme.

In appendix A we include additional basic results on list sample compression schemes. In particular, we prove that any list sample compression scheme generalizes and that learnability is equivalent to *logarithmic*-compressibility. The latter is a variant of the above definition of sample compression where the size of the compression scheme is not a fixed constant d, but rather depends logarithmically on the size of the input sample.

Remark 11 (Functions vs. List Functions) *Let $\mathcal{C}$ be a k-list concept class. Define a function class*

$$\mathcal{F} = \mathcal{F}(\mathcal{C}) = \{f : \mathcal{X} \to \mathcal{Y} : f \prec c \text{ for some } c \in \mathcal{C}\},$$

where $f \prec c$ means $f(x) \in c(x)$ for all $x \in \mathcal{X}$. Note that $\mathcal{F}$ and $\mathcal{C}$ are equivalent in the sense that a sample S is realizable by $\mathcal{F}$ if and only if it is realizable by $\mathcal{C}$. In particular $\mathcal{F}$ is k-learnable (k-compressible) if and only if $\mathcal{C}$ is. Thus, when discussing learnability (or compressibility), we may restrict our attention to (1-list) function classes without losing generality. For this reason in Theorems 1 to 3 we can focus on function classes.

In contrast, this reduction from list-functions to functions does not make sense when studying empirical risk minimization over a k-list concept class $\mathcal{C}$. Moreover, the above reduction from $\mathcal{C}$ to $\mathcal{F}(\mathcal{C})$ does not preserve uniform convergence. Indeed, let $\mathcal{C} = \{c\}$, where c is the 2-list function such that $c(x) = \{0, 1\}$ for all x. Clearly, $\mathcal{C}$ satisfies uniform convergence (because it is finite), however $\mathcal{F}(\mathcal{C}) = \{0, 1\}^{\mathcal{X}}$ does not satisfy uniform convergence when $\mathcal{X}$ is infinite. For this reason in Theorem 4 we focus on k-list function classes.

2.3. Combinatorial Dimensions

We next introduce the Graph and Daniely-Shwartz (DS) dimensions which are known to characterize Uniform Convergence and List PAC learnability.

Definition 12 (Graph Dimension) *A sequence $S = \{x_i\}_{i=1}^n$ is $\mathtt{G}_k$-shattered by a k-list concept class $\mathcal{C}$ if there is $p \in \mathcal{Y}^n$ called a piovt, such that for any $b \in \{0, 1\}^n$ there is $c_b \in \mathcal{C}$ such that $1[p_i \in c_b(x_i)] = b_i$. In other words the (binary) concept class $\{1_{p_i \in c(x_i)} : c \in \mathcal{C}\}$ shatters S in the classical VC sense of shattering. The k-graph dimension of $\mathcal{C}$ is $\mathtt{G}_k(\mathcal{C})$ the size of the largest $\mathtt{G}_k$-shattered sequence, or infinity if there are $\mathtt{G}_k$ shattered sequences of arbitrary size.*

It is well known that the graph dimension characterizes uniform convergence in the following sense

Theorem 13 (Daniely, Sabato, Ben-David, and Shalev-Shwartz (2011)) *A k-list concept class $\mathcal{C}$ satisfies uniform convergence if and only if $\mathtt{G}_k(\mathcal{C}) < \infty$. Furthermore, the uniform convergence rate of $\mathcal{C}$ is $\Theta\big(\sqrt{\frac{\mathtt{G}_k(\mathcal{C})}{n}}\big)$.*

Daniely-Shwartz Dimension. To define the DS dimension we first need to introduce the concept of a pseudo-cube, which is a generalization of the boolean hypercube $\{0, 1\}^n$ when the label space is $\mathcal{Y} = [m]$. We say that $y, y' \in [m]^n$ are neighbors in direction i (or simply i-neighbors) if $y_j = y'_j$ if and only if $j \neq i$. We say that $B \subset \mathcal{Y}^n$ is a pseudo-cube of rank d if each $y \in B$ has at least d distinct neighbors in each direction $i \in [n]$.

Definition 14 (DS Dimension) *A sequence $S = \{x_i\}_{i=1}^n$ is $\mathtt{DS}_k$-shattered by $\mathcal{C}$ if $\{(c(x_1), \ldots, c(x_n)) : c \in \mathcal{C}\}$ contains a pseudo-cube of rank k. The k-DS dimension of $\mathcal{C}$ is $\mathtt{DS}_k(\mathcal{C})$ the size of the largest $\mathtt{DS}_k$ shattered sequences, or infinity if there are $\mathtt{DS}_k$-shattered sequences of arbitrary size.*

As shown by Charikar and Pabbaraju (2022) the $\mathtt{DS}_k$ is the combinatorial dimension that characterizes k-list learnability in the following sense

Theorem 15 (Charikar and Pabbaraju (2022)) *A concept class $\mathcal{C}$ is k-list PAC learnable if and only if $\mathtt{DS}_k(\mathcal{C}) < \infty$.*

3. Technical Overview

In this section, we overview the main ideas which are used in the proofs. We also try to guide the reader on which of our proofs reduce to known arguments and which require new ideas.

3.1. Sample Compression Schemes

Theorems 1, 2, and 3 provide impossibility results for sample compression schemes. These types of results are relatively uncommon in the literature, underscoring the technical challenges involved in comprehensive reasoning about all sample compression schemes. We circumvent these challenges by defining a simpler combinatorial notion of *coverability*, which is implied by compressibility. In essence, if a small-sized sample compression exists, then small covers must also exist, and conversely, the lack of small covers indicates the absence of such a compression scheme.

3.1.1. COVERABILITY

A k-list concept class $\mathcal{H}$ is a k-list cover of a (1-list) function class $\mathcal{C}$ if for any $c \in \mathcal{C}$ there is $h \in \mathcal{H}$ such that $c(x) \in h(x)$ for all $x \in \mathcal{X}$. We say that $\mathcal{C}$ is k-list coverable if there is some polynomial p such that for any finite $S \subset \mathcal{X}$, the finite class $\mathcal{C}|_S = \{c|_S : c \in \mathcal{C}\}$ has a k-list cover $\mathcal{H}$ such that $|\mathcal{H}| \leq p(|S|)$. (Recall that $c|_S$ denotes the restriction (or projection) of the function c to the set S.)

This generalizes the notion of *disambiguation of partial concept classes*: A partial concept class $\mathcal{C}$ is a class of partial functions $c : \mathcal{X} \to \mathcal{Y} \cup \{\star\}$, where $\star \notin \mathcal{Y}$ and $c(x) = \star$ means that c is undefined on x. A class $\mathcal{H}$ is said to disambiguate $\mathcal{C}$ if for every $c \in \mathcal{C}$ there exists $h \in \mathcal{H}$ such that $h(x) = c(x)$ whenever $c(x) \neq \star$. Notice that 1-covers are equivalent to disambiguations. Partial concept classes and disambiguations were studied by (Long, 2001; Attias, Kontorovich, and Mansour, 2022; Alon, Hanneke, Holzman, and Moran, 2021; Hatami, Hosseini, and Meng, 2023; Cheung, Hatami, Hatami, and Hosseini, 2023).

In Lemma 35 we prove that if $\mathcal{C}$ is k-list compressible then it is k-list coverable. Thus, to show that $\mathcal{C}$ is not k-list compressible it suffices to show that it is not k-list coverable. Given this reduction, the proof can be divided into the following steps:

(i) By Alon et al. (2021) there is a partial concept class $\mathcal{C}$ that is 1-list learnable but not 1-list coverable.

(ii) Boosting the hardness of $\mathcal{C}$: by a direct sum argument we show that the k-fold power $\mathcal{C}^k$ is 1-list learnable but not k-list coverable.

(iii.a) Theorems 1 and 2 follow[4] by taking the *minimal disambiguation* of $\mathcal{C}^k$ (defined below).

(iii.b) Theorem 3 follows by taking the *free disambiguation* of $\mathcal{C}^k$ (defined below).

The first two steps yield a partial concept class that is 1-list learnable but not k-list compressible. The next two steps are parallel to each other, these steps employ two types of disambiguations which complete the partial concept class to a total concept class in two ways that witness Theorems 2 and 3.

4. For Theorem 1 we just use the case of $k = 1$.

3.1.2. FREE DISAMBIGUATIONS

Definition 16 (Free Disambiguation) *Let $\mathcal{C}$ be a partial concept class. For each $c \in \mathcal{C}$ let y_c be a distinct new label. Let $\hat{c}$ denote the completion of c such that $\hat{c}(x) = y_c$ whenever $c(x) = \star$. The class $\hat{\mathcal{C}} = \{\hat{c} : c \in \mathcal{C}\}$ is called the free disambiguation of $\mathcal{C}$.*

That is, each function in $\mathcal{C}$ is disambiguated by replacing all instances of $\star$ with a unique label for that function. The following lemma is the key to step (iii.b):

Lemma 17 *A partial concept class $\mathcal{C}$ is k-list learnable (compressible) if and only if its free disambiguation $\hat{\mathcal{C}}$ is k-list learnable (compressible)*

We note that the free disambiguation was used in Pabbaraju (2023) on the partial concept class by Alon et al. (2021) to establish the existence of a learnable class that is not compressible, here we apply it more generally to the classes generated by the direct sum argument.

3.1.3. MINIMAL DISAMBIGUATIONS

Since the free disambiguation invetiably has an infinite label space, it cannot be used to derive Theorems 1 and 2. For that, we introduce a different type of disambiguation:

Definition 18 (Minimal Disambiguation) *Let $\mathcal{C}$ be a partial concept class and let $y_\star$ be a new label. For a partial concept c, let $\bar{c}$ denote the completion of c such that $\bar{c}(x) = y_\star$ whenever $c(x) = \star$. The class $\bar{\mathcal{C}} = \{\bar{c} : c \in \mathcal{C}\}$ is called the minimal disambiguation of $\mathcal{C}$.*

So, all instances of $\star$ are disambiguated by the same new label. Here, the label space of $\bar{\mathcal{C}}$ has just one more label than that of $\mathcal{C}$. In particular, the label space of $\bar{\mathcal{C}}$ is finite whenever the label space of $\mathcal{C}$ is finite. The following lemma is the key to step (iii.a):

Lemma 19 *Let $\mathcal{C}$ be a partial concept class over a finite label space and let $\bar{\mathcal{C}}$ be its minimal disambiguation. Then, (i) if $\mathcal{C}$ is k-list learnable then $\bar{\mathcal{C}}$ is $(k+1)$-list learnable, and (ii) if $\mathcal{C}$ is k-list learnable and $\bar{\mathcal{C}}$ is $(k+1)$-list coverable then $\mathcal{C}$ is k-list coverable.*

The above lemma is significantly more nuanced then Lemma 17. In particular it replaces compressibility with coverability and provides implications in one direction rather than equivalences. Nevertheless it suffices to yield Theorems 1 and 2: we apply it to the basic partial concept class from (Alon et al., 2021) to deduce Theorem 1, and to the classes generated by the direct sum argument to deduce Theorem 2. It would be interesting to find a simpler and more direct argument like in Lemma 17.

3.2. Uniform Convergence

Theorem 4 states an equivalence between learnability, agnostic learnability, and uniform convergence for k-list concept classes $\mathcal{C}$. It is clear that uniform convergence implies agnostic learnability, which implies learnability. Thus, it remains to show that learnability implies uniform convergence. Towards this end it suffices to show that if the graph dimension $\mathtt{G}_k(\mathcal{C})$ (which controls uniform convergence) is unbounded then also the DS dimension $\mathtt{DS}_k(\mathcal{C})$ (which control list learnability) is unbounded.

Assume $S = \{x_i\}_{i=1}^n$ is $\mathtt{G}_k$-shattered by $\mathcal{C}$; thus there exists a pivot $p \in \mathcal{Y}^S$ and functions $\{c_b\}_{b\in\{0,1\}^S}$ such that $p_i \in c_b(x_i)$ if and only if $b_i = 1$. For each c_b let $A_b := \{y \in \mathcal{Y}^S \ : \ \forall i \ y_i \in c_b(x_i)\}$ denote its set of realizable sequences. To lower bound $\mathtt{DS}_k(\mathcal{C})$ we lower bound the size of the union $|\bigcup_{b\in\{0,1\}^n} A_b|$ and apply a version of the Sauer–Shelah-Perles lemma from Charikar and Pabbaraju (2022). Our lower bound on $|\bigcup_{b\in\{0,1\}^n} A_b|$ is based on a coding theoretic approach and consists of the following steps:

1. We first upper bound the size of the intersection $|A_b \cap A_{b'}|$ in terms of the Hamming distance between b and b'.

2. We then utilize the above within an inclusion-exclusion bound on $\{A_b\}_{b \in R}$, where $R \subseteq \{0,1\}^n$ is a random subset of the cube. This yields a lower bound on $|\bigcup_{b \in R} A_b|$ and hence also on $|\bigcup_{b \in \{0,1\}^n} A_b|$.

The first part is just the simple observation that if $y \in c_b(x)$, $y \notin c_{b'}(x)$ then $|c_b(x) \cap c_{b'}(x)| \leq k - 1$. This yields the bound

$$|A_b \cap A_{b'}| \leq \prod_{i=1}^{n} |c_b(x_i) \cap c_{b'}(x_i)| \leq k^n (\frac{k-1}{k})^{d_H(b,b')},$$

where $d_H(b, b') = |\{i \in [n] \ : \ b_i \neq b'_i\}|$ is the Hamming distance between b and b'.

This upper bound on the sizes of the pairwise intersections naturally suggests us to employ the inclusion-exclusion principle to lower bound the size of the entire union. Unfortunately, directly applying it to the entire set $\{A_b\}_{b \in \{0,1\}^n}$ does not work, because the average hamming distance is too small. To overcome this issue we restrict ourselves to a random subset of $\{0, 1\}^n$. Such a set is both large and has a large average Hamming distance.

Acknowledgments

Shay Moran and Tom Waknine are supported by the European Union (ERC, GENERALIZATION, 101039692). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Shay Moran is additionally supported by a Robert J. Shillman Fellowship, by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, and by Israel PBC-VATAT, and by the Technion Center for Machine Learning and Intelligent Systems (MLIS).

References

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of pac learnability of partial concept classes. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 658–671, 2021. URL https://api.semanticscholar.org/CorpusID:236087943.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. *J. Mach. Learn. Res.*, 23:175:1–175:31, 2022. URL http://jmlr.org/papers/v23/20-1353.html.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. page 13, 2022. ISBN 9781665455190. doi: 10.1109/FOCS54457.2022.00093.

Nataly Brukhim, Steve Hanneke, and Shay Moran. Improper multiclass boosting. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 5433–5452. PMLR, 2023. URL https://proceedings.mlr.press/v195/brukhim23a.html.

Moses Charikar and Chirag Pabbaraju. A characterization of list learnability. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, 2022. URL https://api.semanticscholar.org/CorpusID:253420830.

TsunMing Cheung, Hamed Hatami, Pooya Hatami, and Kaave Hosseini. Online learning and disambiguations of partial concept classes. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 42:1–42:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.ICALP.2023.42. URL https://doi.org/10.4230/LIPIcs.ICALP.2023.42.

A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the ERM principle. *Journal of Machine Learning Research*, 16(12):2377–2404, 2015.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. *J. Mach. Learn. Res.*, 16:2377–2404, 2011. URL https://api.semanticscholar.org/CorpusID:12851942.

Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2784–2792, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/59f51fd6937412b7e56ded1ea2470c25-Abstract.html.

Hamed Hatami, Kaave Hosseini, and Xiang Meng. A borsuk-ulam lower bound for sign-rank and its applications. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 463–471. ACM, 2023. doi: 10.1145/3564246.3585210. URL https://doi.org/10.1145/3564246.3585210.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1468–1477, 2015. URL https://api.semanticscholar.org/CorpusID:9657031.

N. Littlestone and M. Warmuth. Relating data compression and learnability. *Unpublished manuscript*, 1986.

Philip M. Long. On agnostic learning with {0, *, 1}-valued and real-valued hypotheses. In David P. Helmbold and Robert C. Williamson, editors, *Computational Learning Theory, 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*, volume 2111 of *Lecture Notes in Computer Science*, pages 289–302. Springer, 2001. doi: 10.1007/3-540-44581-1_19. URL https://doi.org/10.1007/3-540-44581-1_19.

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

Shay Moran, Ohad Sharon, Iska Tsubari, and Sivan Yosebashvili. List online classification. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 1885–1913. PMLR, 2023. URL https://proceedings.mlr.press/v195/moran23a.html.

Chirag Pabbaraju. Multiclass learnability does not imply sample compression, 2023.

Denis Pankratov. Direct sum questions in classical communication complexity. *Master's thesis, University of Chicago*, 2012.

Anup Rao and Amir Yehudayoff. *Bibliography*, page 244–249. Cambridge University Press, 2020.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(90):2635–2670, 2010. URL http://jmlr.org/papers/v11/shalev-shwartz10a.html.

A. Wigderson. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, 2019. ISBN 9780691189130. URL https://books.google.co.il/books?id=-WCqDwAAQBAJ.

Caixia Yan, Minnan Luo, Huan Liu, Zhihui Li, and Qinghua Zheng. Top-k multi-class svm using multiple features. *Information Sciences*, 432:479–494, 2018. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2017.08.004. URL https://www.sciencedirect.com/science/article/pii/S0020025517308642.

Organization

Our appendices are divided into three main sections. In Appendix A we give proofs for results on sample compression. Then in Appendix B the uniform convergence results are proven. Finally in Appendix C we look deeper into the direct sum questions in learning theory and propose some open questions and directions for future research.

Appendix A. Sample Compression Proofs

Here we prove our main results concerning sample compression. We begin with proving equivalence between variable size compressibility and learnability in Section A.1. Then in Section A.2 we prove our main results on sample compression: Theorems 1 to 3.

A.1. Learnability is Equivalent to Variable Size Compressibility

The next definition extends the concept of sample compression schemes to permit the compression size to vary based on the size of the input sample.

Definition 20 (Variable Size Compression) *A k'-list concept class $\mathcal{C}$ is k-list (variable-size) compressible if there exist a sublinear sequence $d(n) = o(n)$ and a reconstruction function $\rho : (\mathcal{X} \times \mathcal{Y})^\star \to \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ such that the following holds. For every sample $S \in (\mathcal{X} \times \mathcal{Y})^n$ there exists $S' = \big((x_1, y_1), \dots (x_{d(n)}, y_{d(n)})\big)$, where $(x_i, y_i) \in S$ for all $i \leq d(n)$ such that*

$$\mathtt{L}_S(\rho(S')) \leq \inf_{c \in \mathcal{C}} \mathtt{L}_S(c). \tag{1}$$

If Equation (1) only holds for $\mathcal{C}$-realizable samples then we say that $\mathcal{C}$ is k-list (variable-size) compressible in the realizable case. As an important special case, we say that $\mathcal{C}$ is compressible of logarithmic size if $d(n) = O(\log n)$.

We state and prove two basic results regarding the connection between compressibility and learnability: (i) an equivalence between learnability and (variable-size) compressibility, and (ii) a quantitative bound on the gnerealization error of sample compression schemes.

Theorem 21 *[Learnability and Logarithmic-Compressibility] Let $k, k' \in \mathbb{N}$. Then, the following statements are equivalent for a k'-list concept class $\mathcal{C}$:*

1. *$\mathcal{C}$ is k-list learnable in the agnostic setting.*
2. *$\mathcal{C}$ is k-list learnable in the realizable setting.*
3. *$\mathcal{C}$ is k-list compressible in the agnostic setting with compression size $d(n) = O(\log n)$.*
4. *$\mathcal{C}$ is k-list compressible in the realizable setting with compression size $d(n) = O(\log n)$.*

Proposition 22 (Compression Size vs. Generalization Error) *Let $k, k' \in \mathbb{N}$, $\mathcal{C}$ be a k'-list concept class that is compressible with compression size $d(n) = O(\log n)$, and set $\mathcal{A} := \rho \circ \kappa$. Then for any realizable distribution $\mathcal{D}$ and $\varepsilon > 0$ we have for all $n > 0$*

$$\Pr\Big(\mathtt{L}_{\mathcal{D}}\big(\mathcal{A}(S)\big) > \varepsilon\Big) \leq 2\exp\big(d(n)\ln(n) - \varepsilon^2 n\big).$$

And for all $n > 0$ the learning curve $\varepsilon_n(\mathcal{A}|\mathcal{D})$ satisfies

$$\varepsilon_n(\mathcal{A}|\mathcal{D}) \leq \sqrt{\frac{(d(n)+1)\ln(n)}{n}} + \frac{2}{n}.$$

The proofs of Theorem 21 and Proposition 22 are adaptations of the proofs of the classical cases of $k = 1$, which can be found e.g. in David et al. (2016). For completeness, we repeat the argument with the necessary adjustments below.

Proof [Proof of Theorem 21 and Proposition 22] We will show that $1 \implies 2 \implies 4 \implies 3 \implies 1$,where the proof of $3 \implies 1$ implies Proposition 22. The direction $1 \implies 2$ is clear by definition. Next, we show that $4 \implies 3$. Indeed let (ρ, κ) be a realizable k-list compression scheme for $\mathcal{C}$. Let $S \subset (\mathcal{X} \times \mathcal{Y})^*$ be some sample and take $S' \subset S$ to be a $\mathcal{C}$ realizable subsample of maximal size. by definition $|S| \cdot \mathtt{L}_S(c) \geq |S \setminus S'|$ for any $c \in \mathcal{C}$. Now since S' is realizable we have $\mathtt{L}_{S'}\big(\rho(\kappa(S'))\big) = 0$ implying

$$|S| \cdot \mathtt{L}_S\Big(\rho\big(\kappa(S')\big)\Big) \leq |S| \cdot \mathtt{L}_{S'}\Big(\rho\big(\kappa(S')\big)\Big) + |S \setminus S'| = |S \setminus S'| \leq |S| \cdot \inf_{c \in \mathcal{C}} \mathtt{L}_S(c).$$

Therefore ρ is also an agnostic k-list reconstruction function for $\mathcal{C}$.

Now we show that $3 \implies 1$. Let ρ be a reconstructor for $\mathcal{C}$ of size $\{d(n)\}_{n=1}^{\infty}$ for some sequence $d(n) = O(\log(n))$. We will show that $\mathcal{A}(S) := \rho\big(\kappa(S)\big)$ is a k-list learner for $\mathcal{C}$. Fix some learning problem $\mathcal{D}$, and draw a sample S of size n according to $\mathcal{D}$. define

$$\mathcal{T} := \{(x_1, y_1), (x_2, y_2), \dots (x_{d(n)}, y_{d(n)}) \ : \ \forall 1 \leq j \leq d(n) \ (x_j, y_j) \in S\}.$$

The set of all possible inputs to our reconstructor (with respect to the sample S). For any $T \in \mathcal{T}$, define

$$\begin{aligned} S_T &= S \setminus T = \{(x, y) \in S \ : (x, y) \notin T\}, \\ h_T &= \rho(T). \end{aligned}$$

For any $s = (x, y) \in S$ define the random variable $X_{s,T} = 1_{y \in h_T(x)}$. So

$$\mathtt{L}_{S_T}(h_T) = \frac{1}{n - d(n)} \sum_{s \in S_T} X_{s,T}.$$

Now since clearly, h_T is independent of S_T and the $\{X_{s,T}\}_{s \in S}$ are independent of each other we may use Hoeffding's inequality to get

$$\Pr(|\mathtt{L}_{S_T}(h_T) - \mathtt{L}_{\mathcal{D}}(h_T)| > \varepsilon) \leq 2e^{-2\varepsilon^2(n - d(n))},$$

by simple union bound deduce

$$\Pr(\exists T \ \text{ s.t } \ |\mathtt{L}_{S_T}(h_T) - \mathtt{L}_{\mathcal{D}}(h_T)| > \varepsilon) \leq 2n^{d(n)} e^{-2\varepsilon^2(n - d(n))} = 2 \exp\big(d(n)(1 + \log n) - 2\varepsilon^2 n\big).$$

Finally, we note that on the above event, we have $|\mathtt{L}_{\mathcal{D}}(\mathcal{A}) - \mathtt{L}_S(\mathcal{A})| \leq \varepsilon$. hence

$$\Pr\big(|\mathtt{L}_{\mathcal{D}}(\mathcal{A}) - \mathtt{L}_S(\mathcal{A})| > \varepsilon\big) \leq 2 \exp\big(d(n)(1 + \log n) - 2\varepsilon^2 n\big).$$

From this, we may use $d(n) = O(\log n)$ (in fact it suffices to assume that $d(n) = o(\frac{n}{\log n})$) to deduce the desired result.

Finally, we show that $2 \implies 4$. Let $\mathcal{A}$ be a k-list learning rule for $\mathcal{C}$. Let $\varepsilon > 0$ be such that $\varepsilon < \frac{1}{2(k+1)}$and take d large enough such that the learning curve $\varepsilon_d(\mathcal{A}|\mathcal{D}) \leq \varepsilon$ for any realizable $\mathcal{D}$. We will show that $\mathcal{C}$ has a compression scheme of size $d(n) = \frac{d \log(3n)}{\varepsilon^2}$, note that ε, d are constants so this indeed gives a logarithmic compression size.

We define the reconstruction function ρ as follows. Given a sample S of size $d \cdot T$ partition it into T samples of size d so $S = (S_1, S_2, \dots, S_T)$, then we can think of $\rho(S)$ as a top k majority vote of the $\{A(S_t)\}_{t=1}^T$, formally for any $x \in \mathcal{X}$ we define $\phi_x : \mathcal{Y} \to \mathbb{N}$, $\Phi_x : \binom{\mathcal{Y}}{k} \to \mathbb{N}$ by

$$\phi_x(y) = |\{t \in [T] \ : \ y \in A(S_t)(x)\}|,$$
$$\Phi_x(Y) = \sum_{y \in Y} \phi_x(y).$$

and then define $\rho(S)(x)$ by

$$\rho(S)(x) = \operatorname*{argmax}_{Y \in \binom{\mathcal{Y}}{k}} \Phi_x(Y),$$

where ties are resolved arbitrarily. Now we need to show that ρ is indeed a reconstruction function for $\mathcal{F}$. Fix some realizable sample $S = \{(x_i, y_i)\}_{i=1}^n$ of size n and define

$$\mathcal{H} := \left\{ \mathcal{A}(S') \ : \ S' \subset \binom{S}{d} \right\}.$$

Note that for any realizable distribution $\mathcal{D}$ over S there is some $h \in \mathcal{H}$ such that $\mathtt{L}_{\mathcal{D}}(h) \leq \varepsilon$. Now let us consider the zero-sum game between a learner and an adversary that goes as follows, the learner picks $h \in \mathcal{H}$ and the adversary picks $(x_i, y_i) \in S$, the learner wins if and only if $y_i \in h(x_i)$. With this view, the above remark states that for any randomized strategy of the adversary, the learner has a deterministic strategy that loses with probability at most ε. Hence by the well-known min-max theorem, there is some randomized strategy, for which the expected loss of the learner is at most ε for any randomized strategy of the adversary. Let μ be the distribution over $\mathcal{H}$ that induce the above strategy and let $\{H_t\}_{t=1}^T$ be T elements of $\mathcal{H}$ independently drawn according to μ where $T = \frac{\log(3n)}{\varepsilon^2}$. We also define

$$X_{t,i} = 1_{y_i \in H_t(x_i)},$$
$$X_i = \frac{1}{T} \sum_{t=1}^T X_{t,i}.$$

note that $EX_{t,i} = EX_i \geq 1 - \varepsilon$ by the choice of μ. Now by Hoeffding's inequality, we have

$$\Pr(|X_i - EX_i| > \varepsilon) \leq 2\exp(-2\varepsilon^2 T) = 2e^{\frac{-2\varepsilon^2 \log(3n)}{\varepsilon^2}} < \frac{1}{n},$$
$$\Pr\left(\bigcup_{i=1}^n |X_i - EX_i| > \varepsilon \right) < 1.$$

Hence, there are some $\{h_t\}_{t=1}^T$ in $\mathcal{H}$ such that

$$\frac{1}{T} |\{t \in [T] \ : \ y_i \in h_t(x_i)\}| > EX_1 - \varepsilon > 1 - 2\varepsilon > \frac{k}{k+1}.$$

Note that $h_t = A(S_t)$ for some $S_t \subset S$, $|S_t| = d$, we claim that $\kappa(S) = (S_1, S_2, \dots S_T)$ will give the desired result. And indeed given some $(x_i, y_i) \in S$ and using the same ϕ_{x_i}, Φ_{x_i} as above we see that $y_i \notin \rho(\kappa(S))$ if and only if there at least k other elements $\{y'_j\}_{i=1}^k$ for which $\phi_{x_i}(y'_j) > \phi_{x_i}(y_i)$. But by the above $\phi_{x_i}(y_i) > \frac{kT}{k+1}$, hence we get

$$kT = \sum_{t=1}^T |h_t(x_i)| = \sum_{y \in \mathcal{Y}} \phi_{x_i}(y) > \sum_{j=1}^k \phi_{x_i}(y'_j) \geq (k+1) \frac{kT}{k+1} = kT.$$

Which is clearly impossible.

■

A.2. Impossibility Results for List Sample Compression

In this section, we prove Theorems 1, 2 and 3. We start by giving technical background and stating some lemmas that are required for our proofs in Section A.2.1. Then, in Section A.2.2 we prove Lemma 33 which is key to the proofs of both Theorems 2 and 3; this lemma uses a direct sum argument to construct a 1-list learnable class that is not k-list coverable. We then utilize this lemma to prove Theorem 2 and Theorem 1 in Section A.2.3 and Theorem 3 in Section A.2.4.

A.2.1. TECHNICAL BACKGROUND

Below we introduce definitions and claims that will be used throughout our proofs in this section.

Sauer-Shelah-Perles Lemma for List Learning. We use the following version of the Sauer-Shelah-Perles (SSP) Lemma by Charikar and Pabbaraju (2022)

Lemma 23 (SSP for DS Dimension (Charikar and Pabbaraju, 2022)) *Let $\mathcal{F} \subset \mathcal{Y}^n$ be a function class with $d = \mathtt{DS}_k(\mathcal{F})$, $m = |\mathcal{Y}|$, then we have*

$$|\mathcal{F}| \leq k^{n-d} \sum_{i=0}^{d} \binom{n}{i} \binom{m}{k+1}^i \leq k^n n^d m^{(k+1)d}.$$

Partial Concepts. We next introduce some useful results from the theory of partial concept classes (Alon et al., 2021).

Definition 24 *Given a domain $\mathcal{X}$, a partial concept c is a function whose domain supp(c) is a subset of $\mathcal{X}$, if supp$(c) = X$ then c is referred to as a total concept. A partial concept class is a collection of partial functions.*

Note that in the above definition, we may take c to be a k-list function. However, for our results, this will not be necessary, as all our partial functions will be of the standard $k = 1$ type.

A standard way to model partial concept classes is to introduce a new label "$\star$" for the inputs on which any function is not defined. In this view, a partial concept class is simply a concept class $\mathcal{C} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$, where for any $c \in \mathcal{C}$, we have $\text{supp}(c) = \{x \in \mathcal{X} \;:\; c(x) \neq \star\} = c^{-1}(\mathcal{Y})$.

The definitions of standard learning theory extend naturally into the partial concept class

Definition 25 *A sample S of size n is realizable by a k-list partial function c if for any $(x, y) \in S$ we have $x \in$ supp(c), $y \in c(x)$. A learning problem $\mathcal{D}$ is realizable by a partial concept class $\mathcal{C} \subset (\binom{\mathcal{Y}}{k} \cup \{\star\})^{\mathcal{X}}$ if for any $n \in \mathbb{N}$ a random sample S of size n drawn independently according to $\mathcal{D}$ is realizable by some $c \in \mathcal{C}$ with probability 1.*

Note that a realizable sequence can not contain the $\star$ symbol in it. Now the notions of compression and learnability can be defined for partial concept classes in the same way as they are defined for total k-list concept classes.

Definition 26 *A partial concept class $\mathcal{C} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$ is k-list covered by a k-list class $\mathcal{H} \subset \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ (and respectively $\mathcal{H}$ is a k-list covering of $\mathcal{C}$) if for any $c \in \mathcal{C}$, there exists $h \in \mathcal{H}$ such that $c(x) \in h(x)$ for all $x \in$ supp(c). In such cases, we write $\mathcal{C} \prec \mathcal{H}$ and $c \prec h$*

Note that in the standard case of $k = 1$, covering is equivalent to disambiguation. Although for $k > 1$ covering can be seen as a generalization of disambiguation, it can also be studied in the context of total k-list classes.

Definition 27 *For any $\mathcal{F} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$, $k \in \mathbb{N}$ we define its k-list covering size*

$$\mathtt{C}_{\mathcal{F}}(n,k) := \sup\{\mathtt{C}(\mathcal{F}|_S, k) \ : \ S \subset \mathcal{X}, \ |S| = n\},$$

where $C(\mathcal{F}|_S, k)$ is the size of the minimal k-list cover of $\mathcal{F}|_S$, i.e.

$$\mathtt{C}(\mathcal{F}|_S, k) := \inf\left\{|\mathcal{H}| \ : \ \mathcal{H} \subset \binom{\mathcal{Y}}{k}^{\mathcal{X}}, \ \mathcal{F}|_S \prec \mathcal{H}\right\}.$$

Direct Sum. We extend the definition of cartesian product to partial concepts.

Definition 28 *Given a k-list partial function $f : \mathcal{U} \to \binom{\mathcal{Y}}{k} \cup \{\star\}$ and a k'-list partial function $g : \mathcal{V} \to \binom{\mathcal{Z}}{k'} \cup \{\star\}$, define their (cartesian) product $f \otimes g : \mathcal{U} \otimes \mathcal{V} \to \binom{\mathcal{Y} \otimes \mathcal{Z}}{kk'} \cup \{\star\}$ as follows*

$$(f \otimes g)(u,v) = \begin{cases} \{(y,z) \ : \ y \in f(u), z \in g(v)\} & u \in \mathit{supp}(f), v \in \mathit{supp}(g), \\ \star & \mathit{else}. \end{cases}$$

Similarly, given a k-list (partial) concept class $\mathcal{F} \subset \binom{\mathcal{Y}}{k}^{\mathcal{U}}$ and a k'-list (partial) concept class $\mathcal{G} \subset \binom{\mathcal{Z}}{k'}^{\mathcal{V}}$, define their product $\mathcal{F} \otimes \mathcal{G}$ as

$$\mathcal{F} \otimes \mathcal{G} = \{f \otimes g \ : \ f \in \mathcal{F}, \ g \in \mathcal{G}\}.$$

We primarily focus on the case where $k = k' = 1$, simplifying the definition to

$$(f \otimes g)(u,v) = \big(f(u), g(v)\big).$$

Definition 29 (Product of Learning Rules) *Given a k-list learning rule $\mathcal{A}_1 : (\mathcal{X}_1 \times \mathcal{Y}_1)^\star \to \binom{\mathcal{Y}_1}{k}^{\mathcal{X}_1}$ and a k'-list learning rule $\mathcal{A}_2 : (\mathcal{X}_2 \times \mathcal{Y}_2)^\star \to \binom{\mathcal{Y}_2}{k'}^{\mathcal{X}_2}$. We define their product $\mathcal{A}_1 \otimes \mathcal{A}_2 : ((\mathcal{X}_1 \otimes \mathcal{X}_2) \times (\mathcal{Y}_1 \otimes \mathcal{Y}_2))^\star \to \binom{\mathcal{Y}_1 \times \mathcal{Y}_2}{k \cdot k'}^{\mathcal{X}_1 \times \mathcal{X}_2}$ as follows*

$$(\mathcal{A}_1 \otimes \mathcal{A}_2)(S) = \mathcal{A}_1(\pi_1(S)) \otimes \mathcal{A}_2(\pi_2(S)),$$

where $\pi_b\big(\{(x_{i,1}, x_{i,2}), (y_{i,1}, y_{i,2})\}_{i=1}^n\big) = \{(x_{i,b}, y_{i,b})\}_{i=1}^n$.

Remark 30 (Products of Learning Rules and Sample Compressions Schemes) *Notice that if $\mathcal{F}$ is k-list compressible with comprssion size d_1 and $\mathcal{G}$ is k'-list compressible with compression size d_2 then $\mathcal{F} \otimes \mathcal{G}$ is kk'-list compressible with compression size $d_1 + d_2$. Moreover we have*

$$\rho\big(\kappa(S)\big) = \rho_1\big(\kappa_1(\pi_1(S))\big) \otimes \rho_2\big(\kappa_2(\pi_2(S))\big).$$

Similarly if $\mathcal{F}$ is k-list learnable and $\mathcal{G}$ is k'-list learnable then $\mathcal{F} \otimes \mathcal{G}$ is kk'-list learnable. Moreover, if $\mathcal{A}_1$ is k-list learner for $\mathcal{F}$ and $\mathcal{A}_2$ is k' list learner for $\mathcal{G}$ then $\mathcal{A}_1 \otimes \mathcal{A}_2$ is kk' list learner for $\mathcal{F} \otimes \mathcal{G}$ satisfying

$$\varepsilon(n|\mathcal{A}_1 \otimes \mathcal{A}_2) \leq \varepsilon(n|A_1) + \varepsilon(n|A_2).$$

A.2.2. MAIN LEMMA: COVERABILITY UNDER DIRECT-SUM

In this part, we prove a main lemma on direct sum of partial concept classes and use it to deduce Lemma 33, establishing the existence of a learnable partial class that is not k-list coverable, which is a key part in the proof of Theorem 3 and Theorem 2.

Lemma 31 (Main Direct Sum Lemma) *Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{U}}$, $\mathcal{G} \subset \mathcal{Z}^{\mathcal{V}}$ be partial concept classes, then for any $k, k', n, n' \in \mathbb{N}$ we have*

$$\min\Big(\mathtt{C}_{\mathcal{F}}(n,k), \mathtt{C}_{\mathcal{G}}(n',k')\Big) \leq \min(n,n') \cdot \mathtt{C}_{\mathcal{F}\otimes\mathcal{G}}(n\cdot n', k+k').$$

In particular, if $\mathcal{F}$ is not k-list coverable and $\mathcal{G}$ is not k'-list coverable then $\mathcal{F} \otimes \mathcal{G}$ is not $(k+k')$-list coverable.

The proof of Lemma 31 requires the following technical claim

Claim 1 *Assume $h : (\mathcal{U} \times \mathcal{V}) \to \binom{\mathcal{Y}\otimes\mathcal{Z}}{k+k'}$ is a $(k+k')$-list function. Then for every $(u,v) \in \mathcal{U} \times \mathcal{V}$ at least one of the following holds:*

- *$\pi_2(h(u,v)) := \{z \in \mathcal{Z} : (y,z) \in h(u,v)$ for some$y \in \mathcal{Y}\}$ has at most k' distinct labels.*
- *For all $z \in \mathcal{Z}$ the set $\{y \in \mathcal{Y} : (y,z) \in h(u,v)\}$ has at most k distinct labels.*

Proof [Proof of claim 1] Let $z' \in Z$, a simple computation gives

$$\begin{aligned} k + k' = |h(u,v)| &= \sum_{z\in\pi_2(h(u,v))} |\{y \in \mathcal{Y} : (y,z) \in h(u,v)\}| \\ &\geq |\{y \in \mathcal{Y} : (y,z') \in h(u,v)\}| + \sum_{\substack{z\in\pi_2(h(u,v)) \\ z\neq z'}} 1 \\ &= |\{y \in \mathcal{Y} : (y,z') \in h(u,v)\}| + |\pi_2(h(u,v))| - 1. \end{aligned}$$

From which the claim follows immediately. ■

Proof [Proof of Lemma 31] Let $U \subset \mathcal{U}$, $V \subset \mathcal{V}$ samples and denote their sizes $|U| = n$, $|V| = n'$. Define $F = \mathcal{F}|_U$, $G = \mathcal{G}|_V$ and note that $F \otimes G = (\mathcal{F} \otimes \mathcal{G})|_{U\otimes V}$, and $|U \otimes V| = n \cdot n'$. Hence there is some H a $(k+k')$ list cover of $F \otimes G$ such that $|H| \leq \mathtt{C}_{\mathcal{F}\otimes\mathcal{G}}(n \cdot n', k+k')$. Without loss of generality we assume $n \leq n'$, so we need to show that either there exists a k-list cover of F of size at most $n \cdot |H|$ or there exists a k'-list cover of G of size at most $n \cdot |H|$.

The proof is based on a win-win argument; we show that if there exists a 'good' $g \in G$ (to be defined below) then we we can cover F using at most $|H|$ k-list functions. And, otherwise, if no $g \in G$ is 'good' then we can cover G using at most $n \cdot |H|$ k'-list functions.

For any $h \in H$, $g \in G$, $v \in \text{supp}(g)$, and $u \in U$ define $h_{g,v}(u)$ by

$$h_{g,v}(u) = \Big\{y \in \mathcal{Y} \ : \ \big(y, g(v)\big) \in h(u,v)\Big\}$$

Now Define g as good if it satisfies

$$(\forall f \in F)(\forall h \in H \text{ such that } f \otimes g \prec h)(\forall u \in \text{supp}(f))(\exists v \in \text{supp}(g)) \ : \ |h_{g,v}(u)| \leq k.$$

Now assume that there is some $g \in G$ that is good, we show how to use it to construct the desired k-list cover of F. Note that if $f \otimes g \prec h$ then $f \prec h_{g,v}$ for all $v \in \text{supp}(g)$. Indeed for any $u \in \text{supp}(f)$ we have

$\big(f(u), g(v)\big) \in h(u, v)$ hence by definition $f(u) \in h_{g,v}$. Now in general the best bound we have on the size of $|h_{g,v}(u)|$ is $(k + k')$, so $h_{g,v}$ may not be a k-list function. To fix this, we wish to 'trim' it somehow, removing some of the unnecessary labels. Note that in the above argument, we got $f(u) \in h_{g,v}(u)$ for all $v \in \text{supp}(g)$, hence it seems natural to take intersection over all such v. For any $h \in H$ we define

$$h_g(u) = \bigcap_{v \in \text{supp}(g)} h_{g,v}(u).$$

We already saw that if $f \otimes g \prec h$ we have that $f(u) \in h_{g,v}$ for all $v \in \text{supp}(g)$. As a result $f(u)$ will be in the intersection and $f \prec h_g$. Now since g for any $f \in F$, $h \in H$ such that $f \otimes g \prec h$ we have that for all $u \in U$ there is some $v \in \text{supp}(g)$ such that $|h_{g,v}(u)| \le k$. In particular, $|h_g(u)| \le k$. So we have that h_g is a k-list function[5] for all $h \in H$ such that there is some $f \in F$ with $f \otimes g \in h$. Hence we may define

$$H_g = \{h_g \ : \ \exists f \in F \ f \otimes g \in h\}.$$

and by the above get that H_g is a k-list cover of F, and clearly $|H_g| \le |H|$.

Now we assume that no $g \in G$ is good and show that this implies the existence of a k'-list cover of G of size at most $n \cdot |H|$. Using De Morgan's laws if no $g \in G$ is good then we have

$$(\forall g \in G)(\exists f \in F)(\exists h \in H \text{ s.t } f \otimes g \prec h)(\exists u \in \text{supp}(f))(\forall v \in \text{supp}(g)) \ : \ |h_{g,v}(u)| > k.$$

Now by Claim 1 if $|h_{g,v}(u)| > k$ then with $\pi_2(y, z) = z$ the projection map we have $|\pi_2\big(h(u, v)\big)| \le k'$. So the above becomes

$$(\forall g \in G)(\exists f \in F)(\exists h \in H \text{ s.t. } f \otimes g \prec h)(\exists u \in \text{supp}(f))(\forall v \in \text{supp}(g)) : |\pi_2\big(h(u, v)\big)| \le k'.$$

Note that if $f \otimes g \prec h$ then for all $u \in \text{supp}(f)$,$v \in \text{supp}(g)$ we have $\big(f(u), g(v)\big) \in h(u, v)$ hence $g(v) \in \pi_2\big(h(u, v)\big)$. Now by the above for all $g \in G$ there are some $f \in F$, $h \in H$, $u \in \text{supp}(f)$ such that $|\pi_2\big(h(u, v)\big)| \le k$ for all $v \in \text{supp}(g)$, but we can't say that $v \to \pi_2\big(h(u, v)\big)$ is a k'-list function since we cant bound its size on $v \notin \text{supp}(g)$. Luckily, when we try to cover g we don't care for the value of the function on $v \notin \text{supp}(g)$, so we may trim it arbitrarily. For any $u \in U$, $h \in H$ let $\binom{h_u : V \to \{\mathcal{Z}}{k'\}}$ be any k'-list function such that $\pi_2\big(h(u, v)\big) \subset h_u(v)$ for any $v \in V$ such that $|\pi_2\big(h(u, v)\big)| \le k'$. Now by the above for any $g \in G$ there is some $h \in H$, $u \in U$ such that $g(v) \in h_u(v)$ for all $v \in \text{supp}(g)$ hence if we define

$$H_U = \{h_u \ : \ h \in H, \ u \in U\}$$

we have that $G \prec H_U$ and clearly $|H_U| \le n \cdot |H|$.

■

Remark 32 *Note that in the second case of the above proof, we get a bound of $n \cdot |H|$ on the size of the cover of G, as opposed to the bound $|H|$ in the case of the cover of F. This n term appears since our functions are partial functions, and when dealing with total classes we may improve that bound. Indeed, similarly to the first case, we may try to trim h_u by taking intersections and define*

$$\pi_h(v) = \bigcap_{u \in U} h_u(v).$$

And while at first, it seems that $\{\pi_h \ : \ h \in H\}$ will give a k'-list cover for G, a closer look will reveal that we don't necessarily have that it covers G since $f \otimes g \prec$ only implies that $g(v) \in h_u(v)$ for $u \in$ supp(f). Hence when we take the intersection over all of U we can no longer guarantee that $g(v) \in \pi_h(v)$, unless supp$(f) = U$, i.e. f is a total function.

5. It is technically possible that the size of $h_v(u)$ is strictly less than k, in which case we add some labels to it arbitrarily to make it a k-list function

Lemma 33 *For any $k \geq 1$ there is a partial concept class $\mathcal{F}_k \subset (\{0,1\}^k \cup \{\star\})^{\mathbb{N}^k}$ such that $\mathcal{F}_k$ is 1-list learnable but $n^{k-1}\mathtt{C}_{\mathcal{F}^k}(n^k, k) \geq n^{\left(\log(n)\right)^{1-o(1)}}$*

Proof The proof is by induction on k. For the base case $k = 1$ we can take $\mathcal{F}_1$ to be the partial class from Alon et al. (2021). It is a learnable binary partial concept class over the natural number, and for any disambiguation of it $\mathcal{C}$ and $n > 0$ there is some $S \subset \mathbb{N}$ of size n such that $|\mathcal{C}|_S| \geq n^{\left(\log(n)\right)^{1-o(1)}}$. Thus it satisfies the required properties.

For the induction step we simply define $\mathcal{F}_{k+1} = \mathcal{F}_k \otimes \mathcal{F}_1$. Indeed $\mathcal{F}_{k+1} \subset (\{0,1\}^k \cup \{\star\})^{\mathbb{N}^k}$ is a 1-list learnable and from the induction assumption and Lemma 31 it satisfies

$$n^{\left(\log(n)\right)^{1-o(1)}} \leq \min\left(\mathtt{C}_{\mathcal{F}}(n,1), n^{k-1}\mathtt{C}_{\mathcal{F}^k}(n^k,k)\right) \leq n^k \mathtt{C}_{\mathcal{F}^{k+1}}(n^{k+1}, k+1).$$

■

A.2.3. PROOF OF THEOREMS 1 AND 2

In this part, we give proof for Theorems 1 and 2. We start by recalling the definitions of the minimal disambiguation and proving some of its basic properties and then use those properties with Lemma 33 to prove those Theorems.

Minimal Disambiguation. Recall the definition of the minimal disambiguation of a partial concept class.

Definition *[Definition 18 restatement] Let $\mathcal{C}$ be a partial concept class and let $y_\star$ be a new label. For a partial concept c, let $\bar{c}$ denote the completion of c such that $\bar{c}(x) = y_\star$ whenever $c(x) = \star$. The class $\bar{\mathcal{C}} = \{\bar{c} : c \in \mathcal{C}\}$ is called the minimal disambiguation of $\mathcal{C}$.*

Note that if $\mathcal{C} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$ is partial concept class that is k-list learnable then its minimal disambiguation $\bar{\mathcal{C}}$ is $k+1$-list learnable. Indeed let $\mathcal{A}$ be a k-list learning rule for $\mathcal{F}$. For any sample $S = \{(x_i, y_i)\}_{i=1}^n$ define $S' = S \setminus (\mathcal{X} \times \{y_\star\})$, note that if S is realizable by $\bar{\mathcal{C}}$ then S' is realizable by $\mathcal{C}$. Now we may define $\bar{\mathcal{A}}$ a $(k+1)$-list learning rule for $\bar{\mathcal{C}}$ by

$$\bar{A}(S)(x) = A(S')(x) \cup \{y_\star\}$$

And one can easily verify that $\bar{A}$ is a learner for $\bar{F}$.

Similarly if $\mathcal{F}$ is k-list compressible with reconstruction function ρ of size k_n we can easily verify that $\bar{\mathcal{F}}$ is $k+1$-list compressible with reconstruction function $\bar{\rho}(S)(x) = \rho(S')(x) \cup \{y_\star\}$. This phenomenon continues to hold in the following useful lemma concerning the covering size function

Lemma 34 *Let $\mathcal{F} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$ be a learnable partial concept class over a finite label space $|\mathcal{Y}| < \infty$. Define $|\mathcal{Y}| = m$, $DS_1(\mathcal{F}) = d$, then we have*

$$\mathtt{C}_{\mathcal{F}}(n,k) \leq (mn)^d \mathtt{C}_{\bar{\mathcal{F}}}(n, k+1)$$

Proof Let $S \subseteq \mathcal{X}$ be finite of size $n = |S|$, let $F = \mathcal{F}|_S$ and $\bar{F} = \bar{\mathcal{F}}|_S$ and let $\mathcal{H}$ be a $(k+1)$-list covering for $\bar{F}$. We need to show that F has a k-list cover of size $(mn)^d|H|$. H is already a cover of F but it consists of $(k+1)$-list functions, thus to construct the desired cover it is enough to replace each $h \in H$ with some 'sufficiently small' set of k-list functions that will cover it. Concretely, for any $h \in \mathcal{H}$ we define

$$\begin{aligned} X_h &= \{x \in S \ : \ y_\star \notin h(x)\} \\ F_h &= \{f|_{X_h} \ : \ f(x) \in h(x) \text{ for all } x \in S\}. \end{aligned}$$

We now introduce the k-list functions that will replace h in the cover of F. For each $y \in \mathcal{Y}$ let A_y be some set of size k such that $y \in A_y$, and for every $f \prec h$ define the k-list function h_f by

$$h_f(x) = \begin{cases} A_{f(x)} & x \in X_h, \\ h(x) \setminus \{y_\star\} & x \notin X_h, \end{cases}$$

Note that h_f is a k-list function: indeed for every $x \in X_h$ the set $A_{f(x)}$ has size k and for $x \notin X_h$, we have that $y_\star \in h(x)$, the size of $h(x)$ is $k+1$, and hence $h(x) \setminus \{y_\star\}$ has size k. Now set

$$H_F = \{h_f \ : \ h \in H, f \in F_h\}.$$

We claim that $F \prec H_F$. Let $f \in F$. Then, since H covers $\bar{F}$ and $\bar{F}$ disambiguates F there must be some $h \in \mathcal{H}$ such that $f \prec h$. We show that $f \prec h_f$: let $x \in \mathrm{supp}(f)$; if $x \notin X_h$ then $f(x) \in h(x) \setminus \{y_\star\} = h_f(x)$. If $x \in X_h$ then $h_f(x) = A_{f(x)}$ and hence contains $f(x)$. To bound the size H_F note that

$$|H_F| = \sum_{h \in H} |\{f|_{X_h} : f \in F, f \prec h\}|.$$

Now notice that $\{f|_{X_h} : f \in F, f \prec h\} \subseteq \{f|_{X_h} : f \in F, \mathrm{supp}(f) \subseteq X_h\}$, because $y_\star \notin h(x)$ whenever $x \in X_h$. Thus, $\{f|_{X_h} : f \in F, f \prec h\}$ is a class of total functions with k-DS dimension at most $\mathtt{DS}_k(\mathcal{F}) = d$ and hence by the Sauer-Shelah-Perles Lemma:

$$|\{f|_{X_h} : f \in F, f \prec h\}| \le |\mathcal{Y}|^d |X_h|^d \le (mn)^d,$$

implying that

$$\mathtt{C}_{\mathcal{F}}(k,n) \le |\mathcal{H}_{\mathcal{G}}| \le \sum_{h \in \mathcal{H}} |\mathcal{G}|_h| \le (mn)^d |\mathcal{H}|.$$

Taking minimum over all possible $\mathcal{H}$ with $\bar{\mathcal{F}} \prec \mathcal{H}$ gives the desired claim. ■

Theorem *[Theorem 1 restatement]There exists a concept class $\mathcal{C}$ over the label space $\mathcal{Y} = \{0,1,2\}$ such that:*

- *$\mathcal{C}$ is 2-list PAC learnable.*
- *$\mathcal{C}$ has no finite 2-list sample compression scheme.*

Theorem *[Theorem 2 restatement] For any $k > 0$ there exists a concept class $\mathcal{C}_k$ over a finite label space $\mathcal{Y}_k$ that satisfies the following:*

1. *$\mathcal{C}_k$ is 2-list PAC learnable.*
2. *$\mathcal{C}_k$ has no finite k-list sample compression scheme.*

Proof [Proof of Theorems 1 and 2] 6Let $\mathcal{F}_k$ be the partial concept class given by Lemma 33, so $\mathcal{F}_k$ is learnable and $n^k \mathtt{C}_{\mathcal{F}_k}(n^k, k) \ge n^{\left(\log(n)\right)^{1-o(1)}}$. Let $\mathcal{C}_k = \bar{\mathcal{F}}_k$ be the minimal disambiguation of $\mathcal{F}_k$. Note that $\bar{\mathcal{F}}_k$ is 2-list learnable since $\mathcal{F}_k$ is 1-list learnable. Now let $d = \mathtt{DS}_k(\mathcal{F}_k)$ and m be the size of the label space of $\mathcal{F}_k$, then by Lemma 34 we have

$$n^k (mn^k)^d \mathtt{C}_{\bar{\mathcal{F}}_k}(n^k, k+1) \ge n^k \mathtt{C}_{\mathcal{F}_k}(n^k, k) \ge n^{\left(\log(n)\right)^{1-o(1)}}.$$

for all $n > 0$ large enough. Hence the function $n \to \mathtt{C}_{\bar{\mathcal{F}}_k}(n, k+1)$ is not polynomial in n, and since the label space of $\bar{\mathcal{F}}_k$ is finite we may use the above with Lemma 35 to deduce that $\bar{\mathcal{F}}_k$ is not $(k+1)$-list compressible. We note that the case $k = 1$ gives Theorem 1, where we can look at the proof of 33 to see that the label space of $\mathcal{F}_1$ will be of size 2 hence the size of the label space of $\mathcal{C}_1 = \bar{\mathcal{F}}_1$ will be 3. ■

A.2.4. PROOF OF THEOREM 3

Free Disambiguation. Recall the definition of the free disambiguation of a partial concept class.

Definition *[Definition 16 restatement] Let $\mathcal{C}$ be a partial concept class. For each $c \in \mathcal{C}$ let y_c be a distinct new label. Let $\hat{c}$ denote the completion of c such that $\hat{c}(x) = y_c$ whenever $c(x) = \star$. The class $\hat{\mathcal{C}} = \{\hat{c} : c \in \mathcal{C}\}$ is called the free disambiguation of $\mathcal{C}$.*

One can easily verify that for any partial concept class $\mathcal{C} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$ its free disimbaguation $\hat{\mathcal{C}}$ is k-list learnable (compressible) if and only if $\mathcal{C}$ is k-list learnable (compressible). Finally, we prove a short lemma that relates compressibility to polynomial growth of the covering size function.

Lemma 35 *For any partial concept class $\mathcal{F} \subset (\mathcal{Y} \cup \{\star\})^{\mathcal{X}}$ that is k-list compressible with a compression scheme of size d we have $\mathtt{C}_{\mathcal{F}}(n, k) \leq |\mathcal{Y}|^d n^d$ for all $n > 0$.*

Proof Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be such class and without loss of generality assume that $|\mathcal{X}| = n$ (else we look on $\mathcal{F}|_S$ for some arbitrary $S \subset \mathcal{X}$ of size n). Let ρ be a k-list reconstructor of size d, so for any $f \in \mathcal{F}$ we have some $S \in (\mathcal{X} \times \mathcal{Y})^d$ with $f \prec \rho(S)$. Hence we can define

$$\mathcal{H} = \{\rho\big(S\big) \ : \ S \in (\mathcal{X} \times \mathcal{Y})^d\}$$

And get that $\mathcal{F} \prec \mathcal{H}$, and clearly $|\mathcal{H}| \leq (|\mathcal{X}| \cdot |\mathcal{Y}|)^d = (n \cdot |\mathcal{Y}|)^d$. ∎

Proof [Proof of Theorem 3] Let $\mathcal{F}_k$ be the partial concept class given by Lemma 33, so $\mathcal{F}_k$ is learnable and $n^k \mathtt{C}_{\mathcal{F}_k}(n^k, k) \geq n^{\left(\log(n)\right)^{1-o(1)}}$. Hence by Lemma 35 we have that $\mathcal{F}_k$ has no compression scheme of constant size d for any $d > 0$. Now $\mathcal{C}_k = \hat{\mathcal{F}}_k$ is the free disambiguation of $\mathcal{F}_k$, and since $\mathcal{F}_k$ is learnable but not k-list compressible so is $\mathcal{C}_k$. ∎

Appendix B. Uniform Convergence Proofs

In this section we prove Theorem 4 and its quantitative parallel Theorem 37.

We start by proving Lemma 36 which is key to the proof. This lemma relates the graph dimension of a class to the number of sequences realizable by it. Then the theorem will follow by an application of the Sauer–Shelah-Perles lemma for list functions.

Lemma 36 *Given a k-list class $\mathcal{C} \subset \binom{\mathcal{Y}}{k}^{\mathcal{X}}$ that $\mathtt{G}_k$ shatters $S \in \mathcal{X}^n$, we let $p \in \mathcal{Y}^n$ be a pivot and let $\{c_b\}_{b \in \{0,1\}^n} \subset \mathcal{C}|_S$ be witnesses for the shattering, such that $p_i \in c_b(x_i)$ if and only if $b_i = 1$. Denote by $A_b = \{y \in \mathcal{Y}^n \ : \ \forall i \ \ y_i \in c_b(x_i)\}$, the collection of c_b realizable functions. Then we have*

$$|\bigcup_{b \in \{0,1\}^n} A_b| \geq \frac{(2k)^n}{4(2k-1)^n} k^n.$$

Proof Clearly if $y \in A_b \cap A_{b'}$ then $y_i \in c_b(x_i) \cap c_{b'}(x_i)$ hence

$$|A_b \cap A_{b'}| \leq \prod_{i=1}^{n} |c_b(x_i) \cap c_{b'}(x_i)|.$$

Now in general we can only know that $|c_b(x_i) \cap c_{b'}(x_i)| \leq k$ but if $b_i \neq b'_i$ then p_i will be in exactly one of $c_b(x_i), c_{b'}$. Hence in that case we have $|c_b(x_i) \cap c_{b'}| \leq k-1$. Now if we denote d_H the Hamming distance $d_H(b,b') = |\{i \in [n] \ : \ b_i \neq b'_i\}|$ we can use the above to deduce that

$$|A_b \cap A_{b'}| \leq \prod_{b_i = b'_i} k \cdot \prod_{b_i \neq b'_i} (k-1) = k^{n-d_H(b,b')}(k-1)^{d_H(b,b')} = k^n (\frac{k-1}{k})^{d_H(b,b')}.$$

Now we note that for any $I \subset \{0,1\}^n$ we have by the inclusion-exclusion principle

$$|\bigcup_{b\in\{0,1\}^n} A_b| \geq |\bigcup_{b\in I} A_b| \geq \sum_{b\in I}|A_b| - \sum_{b,b'\in I}|A_b \cap A_{b'}| \geq k^n|I| - \sum_{b,b'\in I} k^n(\frac{k-1}{k})^{d_H(b,b')}.$$

So to bound our desired expression we just need to find a large $I \subset \{0,1\}^n$ with a large average hamming distance, this is a standard problem in coding theory, and in our case, a probabilistic approach will suffice. Fix $m > 0$ a constant that will be chosen later and pick m elements from $\{0,1\}^n$ independently and uniformly, denote them by $I = \{B_i\}_{i=1}^m$. Note that for any $B, B' \in I$ we have that $d_H(B,B') = \sum_{i=1}^n 1_{B_i \neq B'_i}$ is a sum of independent Bernoulli random variables, hence by simple computation we have

$$\mathbb{E}\left(\frac{k-1}{k}\right)^{d_H(B,B')} = \prod_{i=1}^n \mathbb{E}\left(\frac{k-1}{k}\right)^{1_{B_i\neq B'_i}} = \prod_{i=1}^n \left(\frac{1}{2} + \frac{k-1}{2k}\right) = \frac{(2k-1)^n}{(2k)^n}.$$

Form which we get

$$\mathbb{E}\Big[\sum_{B,B'\in I} k^n(\frac{k-1}{k})^{d_H(B,B')}\Big] = k^n\binom{m}{2}\frac{(2k-1)^n}{(2k)^n} \leq m^2 k^n \frac{(2k-1)^n}{(2k)^n}.$$

Now putting this back into the inequality we calculated above we can see that

$$|\bigcup_{b\in\{0,1\}^n} A_b| \geq mk^n - m^2k^n\frac{(2k-1)^n}{(2k)^n}$$

Now a simple calculation will reveal that the maximal value is attained when $m = \frac{(2k)^n}{2(2k-1)^n}$ for which we get the desired inequality

$$|\bigcup_{b\in\{0,1\}^n} A_b| \geq \frac{(2k)^n}{4(2k-1)^n}k^n.$$

■

Theorem 37 *Let $\mathcal{C}$ be some k-list concept class. Let $d = \mathtt{DS}_k(\mathcal{C})$, $g = \mathtt{G}_k(\mathcal{C})$ and $m = |\mathcal{Y}|$ the size of the label space of $\mathcal{C}$. Then we have*

$$4g^d m^{(k+1)d} \geq (\frac{2k}{2k-1})^g.$$

In particular if $g = \infty$ then so is d as the above can be simplified to

$$d = \tilde{\Omega}\Big(\frac{g}{k\cdot \ln(g\cdot m)}\Big).$$

Proof [Proof of Theorem 4 and 37] Let $S \in \mathcal{X}^g$ be a sample of size g that is $\mathtt{G}_k$ shattered by $\mathcal{C}$. Define $\mathcal{F} = \mathcal{F}(\mathcal{C}|_S) = \{f \in \mathcal{Y}^g \ : \ \exists h \in \mathcal{C}\ ,\ f \prec h|_S\}$. By Lemma 36 we have that $|\mathcal{F}| \geq \frac{(2k)^g}{4(2k-1)^g}k^g$, while by Lemma 23 we have that $|\mathcal{F}| \leq k^g(gm^{k+1})^{\mathtt{DS}_k(\mathcal{F})}$. Now we note that $\mathtt{DS}_k(\mathcal{C}|_S) = \mathtt{DS}_k(\mathcal{F})$ since a sample is $\mathcal{C}|_S$ realizable if and only if it is $\mathcal{F}$ realizable, from which we may deduce the desired result.

■

Appendix C. Direct Sums and Open Questions

Below we introduce some open questions and research directions that arise from our study of direct sums. We start by focusing on studying how learning and uniform convergence rate scale under direct sum. We then ask similar types of questions for other learning resources as well as on different combinatorial parameters that arise in learning theory.

C.1. Direct Sum of Learning Rates

One of the most natural questions regarding direct sums of learning problems is the following question: given two learning tasks, can we learn both of them in a faster way than learning each individuality? Perhaps the simplest case is of multiple instances of the same learning task. Let $\mathcal{C}$ be a concept class and recall that for $r \in \mathbb{N}$, the r'th power of $\mathcal{C}$ is denoted by $\mathcal{C}^r = \underbrace{\mathcal{C} \otimes \mathcal{C} \cdots \otimes \mathcal{C}}_{r \text{ times}}$.

How does the learning rate of $\mathcal{C}^r$ scale in terms of the learning rate of $\mathcal{C}$?

This problem can be investigated with respect to various formulations of 'learning rate', for example:

Open Question *[Open question 1 restatement] Let $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and let $\varepsilon(n|\mathcal{C})$ denote the realizable PAC learning curve of $\mathcal{C}$ (see Definition 6). For $r \in \mathbb{N}$ let $\mathcal{C}^r = \Pi_{i=1}^r \mathcal{C}$ be the r-fold cartesian power of $\mathcal{C}$. By a union bound, learning each component independently gives*

$$\varepsilon(n|\mathcal{C}^r) \leq r \cdot \varepsilon(n|\mathcal{C}).$$

Can the upper bound be asymptotically improved for some classes $\mathcal{C}$?

Another natural version of the above is assuming a fixed marginal distribution $\mathcal{D}$:

Definition 38 *Let $\mathcal{D}$ be a fixed distribution over the domain $\mathcal{X}$ and let $\mathcal{C}$ be a concept class. For any $c : \mathcal{X} \to \mathcal{Y}$ let $\mathcal{D}_c$ be the distribution in which $(x, y) \sim \mathcal{D}_c$ satisfies $x \sim \mathcal{D}$ and $y = c(x)$. Define the fixed-marginal learning curve $\varepsilon(n|\mathcal{D}, \mathcal{C})$ by*

$$\varepsilon(n|\mathcal{D}, \mathcal{C}) = \inf_{\mathcal{A}} \sup_{c \in \mathcal{C}} \varepsilon_n(\mathcal{D}_c|\mathcal{A})$$

where the infimum is taken over all learning rules $\mathcal{A}$.

Note that for any $c \in \mathcal{C}$ we have that $\mathcal{D}_c$ is a realizable distribution, hence $\varepsilon(n|\mathcal{D}, \mathcal{C}) \leq \varepsilon(n|\mathcal{C})$ always holds.

For any $r > 0$, let $\mathcal{D}^r$ be the product measure over $\mathcal{X}^r$.

Open Question 2 *Similarly to the case of the PAC learning curve, a simple union bound will give the upper bound*

$$\varepsilon(n|\mathcal{D}^r, \mathcal{C}^r) \leq r \cdot \varepsilon(n|\mathcal{D}, \mathcal{C}).$$

Can the upper bound be asymptotically improved for some concept classes $\mathcal{C}$ and marginal distributions $\mathcal{D}$?

One can ask similar questions about agnostic learning curves and uniform convergence. However, in these cases the baseline additive upper bound does not apply. The reason is because these curves concern relative quantities (indeed, the agnostic learning curve measures the excess loss and uniform convergence curve measures the maximum difference between the empirical and population losses).

For instance, given a distribution $\mathcal{D}$ over the product space $(\mathcal{X}^2 \times \mathcal{Y}^2)$ defined with marginal distributions $\mathcal{D}_1, \mathcal{D}_2$ over $(\mathcal{X} \times \mathcal{Y})$ we have by the union bound that $\mathrm{L}_{\mathcal{D}}(h_1 \otimes h_2) \leq \mathrm{L}_{\mathcal{D}_1}(h_1) + \mathrm{L}_{\mathcal{D}_2}(h_2)$. Similarly if $S = \{((x_{i,1}, x_{i,2}), (y_{i,1}, y_{i,2}))\}_{i=1}^n$, letting $S_b = \{(x_{i,b}, y_{i,b})\}_{i=1}^n$, we have $\mathrm{L}_S(h_1 \otimes h_2) \leq \mathrm{L}_{S_1}(h_1) + \mathrm{L}_{S_2}(h_2)$. These bounds, however, do not allow us to bound the *difference* $|\mathrm{L}_D(h_1 \otimes h_2) - \mathrm{L}_S(h_1 \otimes h_2)|$ as needed to bound the uniform convergence rate.

Open Question 3 *Let $\varepsilon_{\texttt{UC}}(n|\mathcal{C})$ be the uniform convergence curve of $\mathcal{C}$ (Definition 9). How does $\varepsilon_{\texttt{UC}}(n|\mathcal{C}^r)$ scale as a function of $\varepsilon_{\texttt{UC}}(n|\mathcal{C})$ and r?*

A similar phenomenon happens in the case of agnostic learning: define the agnostic learning curve of a concept class $\mathcal{C}$ by

$$\varepsilon_{\texttt{agn}}(n|\mathcal{C}) = \inf_{\mathcal{A}} \sup_{\mathcal{D}} (\mathtt{L}_{\mathcal{D},n}(\mathcal{A}) - \mathtt{L}_{\mathcal{D}}(\mathcal{C})),$$

where the infimum is taken over all learning rules $\mathcal{A}$, the supremum is taken over all distributions, and $\mathtt{L}_{\mathcal{D},n}(\mathcal{A}) = \mathbb{E}_{S\sim\mathcal{D}^n}[\mathtt{L}_{\mathcal{D}}(\mathcal{A}(S))]$, and $\mathtt{L}_{\mathcal{D}}(\mathcal{C}) = \inf_{c\in\mathcal{C}} \mathtt{L}_{\mathcal{D}}(c)$. Here again we do not have simple bounds to the agnostic learning curve of $\mathcal{C}^r$ in terms of the agnostic learning curve of $\mathcal{C}$.

Open Question 4 *Let $\varepsilon_{\texttt{agn}}(n|\mathcal{C})$ be the agnostic PAC learning curve of $\mathcal{C}$. How does $\varepsilon_{\texttt{agn}}(n|\mathcal{C}^r)$ scale as a function of $\varepsilon_{\texttt{agn}}(n|\mathcal{C})$ and r?*

C.2. Direct Sum of Learnability Parameters

Another important resource in the context of list learning is the minimal list size k for which a given class $\mathcal{C}$ is k-list PAC learnable. This raises the following questions:

Open Question 5 *Let $\mathcal{C}_1, \mathcal{C}_2$ be concept classes and assume k_1 and k_2 are the minimal integers such that $\mathcal{C}_1$ is k_1-list PAC learnable and $\mathcal{C}_2$ is k_2-list PAC learnable. What is the minimal integer k such that $\mathcal{C}_1 \otimes \mathcal{C}_2$ is k-list PAC learnable? How does it scale as a function of k_1 and k_2.*

It is not hard to see that $k \leq k_1 \cdot k_2$ by just learning each component independently and taking all possible pairs of labels in the marginal lists. We also show that $k \geq (k_1 - 1) \cdot (k_2 - 1)$ (see Equation (2) below). However, it remains open to determine tight bounds on k.

We raise the parallel question regarding compressibility:

Open Question 6 *Let $\mathcal{C}_1, \mathcal{C}_2$ be concept classes and assume k_1 and k_2 are the minimal integers such that $\mathcal{C}_1$ is k_1-list compressible and $\mathcal{C}_2$ is k_2-list compressible. What is the minimal integer k such that $\mathcal{C}_1 \otimes \mathcal{C}_2$ is k-list compressible? How does it scale as a function of k_1 and k_2.*

A natural way to approach questions such as the ones above and in Section C.1 is by analyzing combinatorial parameters that capture the corresponding resources.

Open Question 7 *Let $\mathcal{F}, \mathcal{G}$ be concept classes, and let $\mathtt{dim}(\cdot)$ be either the Graph dimension, the Natarajan dimension, the Littlestone dimension, or the Daniely-Shwartz dimension. How does $\mathtt{dim}(\mathcal{F} \otimes \mathcal{G})$ scale in terms of $\mathtt{dim}(\mathcal{F})$ and $\mathtt{dim}(\mathcal{G})$?*

We next provide some preliminary results.

Proposition 39 *Let $\mathtt{d_N}(\cdot)$ be the Natarajan dimension, and let $\mathcal{F}$ and $\mathcal{G}$ be concept classes. Then,*

$$\mathtt{d_N}(\mathcal{F}) + \mathtt{d_N}(\mathcal{G}) - 2 \leq \mathtt{d_N}(\mathcal{F} \otimes \mathcal{G}) \leq \mathtt{d_N}(\mathcal{F}) + \mathtt{d_N}(\mathcal{G}).$$

Proof Let $U = \{u_i\}_{i=1}^n$, $V = \{v_j\}_{j=1}^m$ be sets that are Natarajan shattered by $\mathcal{F}$ and $\mathcal{G}$ respectively so $\mathcal{F}|_U$ and $\mathcal{G}|_V$ contains a Cartesian product of the form

$$\prod_{i=1}^{n} \{f_i, \tilde{f}_i\} \subset \mathcal{F}|_U,$$
$$\prod_{j=1}^{m} \{g_j, \tilde{g}_j\} \subset \mathcal{G}|_V,$$

where $f_i \neq \tilde{f}_i$, $g_j \neq \tilde{g}_j$ for all i, j. Now define $S = \{(u_i, v_1)\}_{i=2}^{n} \cup \{(u_1, v_j)\}_{j=2}^{m}$ and note that $|S| = |U| + |V| - 2$. We claim that S is Natarajan shattered by $\mathcal{F} \otimes \mathcal{G}$. Indeed,

$$\prod_{i=2}^{n}\{(f_i, g_1), (\tilde{f}_i, g_1)\} \times \prod_{j=2}^{m}\{(f_1, g_j), (f_1, \tilde{g}_j)\} \subseteq (\mathcal{F} \otimes \mathcal{G})|_S.$$

Taking supremum over all such U, V we get that $\mathtt{d_N}(\mathcal{F}) + \mathtt{d_N}(\mathcal{G}) - 2 \leq \mathtt{d_N}(\mathcal{F} \otimes \mathcal{G})$. Now for the other part of the inequality let $S = \{(u_i, v_i)\}_{i=1}^{n}$ be some set that is Natrajan shattered by $\mathcal{F} \otimes \mathcal{G}$, meaning that

$$\prod_{i=1}^{n}\{(f_i, g_i), (\tilde{f}_i, \tilde{g}_i)\} \subset \mathcal{F} \otimes \mathcal{G}|_S.$$

Where $(f_i, g_i) \neq (\tilde{f}_i, \tilde{g}_i)$ for all i, but we can have $f_i = \tilde{f}_i$ or $g_i = \tilde{g}_i$. Now we can define

$$U = \{u_i \ : \ f_i \neq \tilde{f}_i\}.V = \{v_i \ : \ g_i \neq \tilde{g}_i\}.$$

Note that since $(f_i, g_i) \neq (\tilde{f}_i, \tilde{g}_i)$ we have that $|U| + |V| \geq n$ and we claim that U and V are Natarajan shattered by $\mathcal{F}$ and $\mathcal{G}$ respectively. And indeed clearly we have

$$\prod_{u_i \in U}\{(f_i, \tilde{f}_i\} \subset \mathcal{F}|_U.$$
$$\prod_{v_i \in V}\{g_i, \tilde{g}_i\} \subset \mathcal{G}|_V.$$

By definition of U, V we have $f_i \neq \tilde{f}_i$ for all i such that $u_i \in U$ and $g_i \neq \tilde{g}_i$ for all i such that $v_i \in V$. Hence we have

$$|S| \leq |U| + |V| \leq \mathtt{d_N}(\mathcal{F}) + \mathtt{d_N}(\mathcal{G}).$$

Taking supermum over all such S will give the desired result. ■

Proposition 40 *Let* $\mathtt{LS}(\cdot)$ *be the Littlestone dimension, then for any* $\mathcal{F}, \mathcal{G}$ *concept classes we have*

$$\mathtt{LS}(\mathcal{F} \otimes \mathcal{G}) = \mathtt{LS}(\mathcal{F}) + \mathtt{LS}(\mathcal{G})$$

Proof We use the fact that the Littlestone dimension equals the optimal mistake bound in deterministic online learning Littlestone and Warmuth (1986). Note that if $\mathcal{A}_1$ is an online learner that makes at most n_1 mistakes on $\mathcal{F}$ and $\mathcal{A}_2$ is an online learner that makes at most n_2 mistakes on $\mathcal{G}$ then $\mathcal{A}_1 \otimes \mathcal{A}_2$ is an online learner that makes at most $n_1 + n_2$ mistakes on $\mathcal{F} \otimes \mathcal{G}$. Hence using the equivalence between Littlestone dimension and mistake bound we deduce $\mathtt{LS}(\mathcal{F} \otimes \mathcal{G}) \leq \mathtt{LS}(\mathcal{F}) + \mathtt{LS}(\mathcal{G})$.

Now for the other direction let $T_1 = (V_1, E_1)$ be a binary tree that is Littlestone shattered by $\mathcal{F}$ and let $T_2 = (V_2, E_3)$ be a binary tree that is Littlestone shattered by $\mathcal{G}$. We will use T_1, T_2 to construct a binary tree T that is shattered by $\mathcal{F} \otimes \mathcal{G}$, and its depth is the sum of the depths of T_1 and T_2. Our approach is to take a copy of T_1 and concatenate a copy of T_2 to each of its leaves, modifying the vertices as necessary. Formally, let r_2 be the root of T_2 and L_1 be the set of leaves of T_1 and define $T = (V, E)$ by

$$V = \{(v, r_2) \ : v \in V_1\} \cup \{(l, v) \ : v \in V_2, \ l \in L_1\}$$
$$E = \{(v, r_2) \to (u, r_2) \ : v \to u \in E_1\} \cup \{(l, v) \to (l, u) \ : \ l \in L_1, \ v \to u \in E_2\}$$

So each path σ in T is of the form

$$(r_1, r_2) \to (v_1, r_2) \cdots \to (v_n, r_2) \to (v_n, u_1) \cdots \to (v_n, u_m),$$

where $\sigma_1 = r_1 \to v_1, \cdots \to v_n$ is a path in T_1 and $\sigma_2 = r_2 \to u_1 \cdots \to u_n$ is a path in T_2. Hence if $f \in \mathcal{F}$, $g \in \mathcal{G}$ are the functions that realize the paths σ_1, σ_2 then $f \otimes g$ realize σ, implying that T is Littelstone shattered by $\mathcal{F} \otimes \mathcal{G}$, and we can easily verify that $d(T) = d(T_1) + d(T_2)$ where $d(\cdot)$ denote the depth of a binary tree. ■

Proposition 41 *Let $\mathcal{F}, \mathcal{G}$ be partial function classes, and let $k, k' \geq 1$. Then we have the following*

1. $\mathtt{DS}_{k \cdot k'}(\mathcal{F} \otimes \mathcal{G}) \geq \min\big(\mathtt{DS}_k(\mathcal{F}), \mathtt{DS}_{k'}(\mathcal{G})\big)$.
2. $\mathtt{DS}_{\min(k,k')}(\mathcal{F} \otimes \mathcal{G}) \geq \mathtt{DS}_k(\mathcal{F}) + \mathtt{DS}_{k'}(\mathcal{G}) - 1$.

Proof Let $U = \{u_i\}_{i=1}^n$ be a set that is $\mathtt{DS}_k$ shattered by $\mathcal{F}$ and let $V = \{v_i\}_{i=1}^m$ be a set that is $\mathtt{DS}_{k'}$ shattered by $\mathcal{G}$. Without loss of generality assume that $n \leq m$. Define $S = \{(u_i, v_i)\}_{i=1}^n$ and notice that if f is a neighbor of f' in the u_i direction and g is a neighbor of g' in v_i direction then $f \otimes g$ is a neighbor of $f' \otimes g'$ in the (u_i, v_i) direction. Since each $f \in \mathcal{F}|_U$ has k neighbors in each direction, and each $g \in \mathcal{G}|_V$ has k' neighbors in each direction we have that each $f \otimes g \in \mathcal{F}|_U \otimes \mathcal{G}|_S$ has kk' neighbors in each direction, implying the first claim.

For the second claim, set $T = \{(u_1, v_i)\}_{i=2}^n \cup \{(u_i, v_1)\}_{i=2}^m \cup \{(u_1, v_1)\}$. In a similar way to the above, now $f \otimes g$ is a neighbor of $f' \otimes g$ in the (u_i, v_1) direction and of $f \otimes g'$ in the (u_1, v_i) direction. So each $f \otimes g \in \mathcal{F} \otimes \mathcal{G}|_T$ has k neighbors in the (u_i, v_1) direction and k' neighbors in the (u_1, v_j) direction for each $1 \leq i \leq n$, $1 \leq j \leq m$. From which the second claim follows.

■

Note that Proposition 41 has direct implications relevant to Open Question 5. Specifically, it implies that if $\mathcal{F}$ is not k-list learnable and $\mathcal{G}$ is not k'-list learnable then $\mathcal{F} \otimes \mathcal{G}$ is not $k \cdot k'$-list learnable. Conversely we know that if $\mathcal{F}$ is k-list learnable and $\mathcal{G}$ is k' list learnable then $\mathcal{F} \otimes \mathcal{G}$ is $k \cdot k'$-list learnable. Thus, letting $K(\mathcal{C})$ denote the minimal k such that a concept class $\mathcal{C}$ is k-list learnable (or infinity if there is no such k) we can summarize the above as

$$(K(\mathcal{F}) - 1)(K(\mathcal{G}) - 1) \leq K(\mathcal{F} \otimes \mathcal{G}) \leq K(\mathcal{F}) \cdot K(\mathcal{G}) \tag{2}$$

Note that this result can be seen as parallel with Lemma 31 which implies that if $\mathcal{F}$ is not k-list coverable and $\mathcal{G}$ is not k'-list coverable then $\mathcal{F} \otimes \mathcal{G}$ is not $(k + k')$-list coverable. It will be interesting to sharpen the latter result, replacing the $(k + k')$ term by $k \cdot k'$ like in Equation 2 above. We may also ask similar questions about compressibility, an answer to which would be relevant to Open Question 6.

Proposition 41 also has implications to Open Question 1 for (1-list) PAC learnable classes. Indeed, let $\mathcal{C}$ be a PAC learnable class. Then, by Item 2 in Proposition 41, it follows that $\mathtt{DS}_1(\mathcal{C}^r) \geq r\mathtt{DS}_1(\mathcal{C})$. Hence, since the Daniely-Shwartz dimension lower bounds the PAC learning curve (Charikar and Pabbaraju, 2022) we get

$$\varepsilon(n|\mathcal{C}^r) \geq \frac{\mathtt{DS}_1(\mathcal{C}^r)}{n} \geq \frac{r \cdot \mathtt{DS}_1(\mathcal{C}) - r}{n}.$$

Thus, if it turns out that the realizable PAC learning curve is $\Theta\big(\frac{\mathtt{DS}_1}{n}\big)$ then the above in combination with the naive union bound argument mentioned in Open Question 1 would answer this question up to universal multiplicative constants.