Finite and Confident Teaching in Expectation: Sampling from Infinite Concept Classes

Jose Hernández-Orallo[1] and **Jan Arne Telle**[2]

Abstract. We investigate the teaching of infinite concept classes through the effect of the *learning* prior (which is used by the learner to derive posteriors giving preference of some concepts over others and by the teacher to devise the teaching examples) and the *sampling* prior (which determines how the concepts are sampled from the class). We analyse two important classes: Turing machines and finite-state machines. We derive bounds for the teaching dimension when the learning prior is derived from a complexity measure (Kolmogorov complexity and minimal number of states respectively) and analyse the sampling distributions that lead to finite expected teaching dimensions. The learning prior goes beyond a complexity or preference choice when we use it to increase the confidence of identification, expressed as a posterior, which increases as more examples are given. We highlight the existing trade-off between three elements: the bound on teaching dimension, the representativeness of the sample and the certainty of the identification. This has implications for the understanding of what teaching from rich concept classes to machines (and humans) entails.

1 Introduction

Learning from examples when the concept class is rich and infinite is commonly considered a very hard computational problem. Positive results in theory and practice usually assume an infinite but not very expressible class, or a strong bias, usually as a prior distribution over the concept class. A uniform choice for this distribution for discrete concept classes leads to zero probabilities or, worse, to no-free lunch results [38, 39]. Consequently, other biases are usually assumed, either related to the application problem at hand or based on some notion of resources used by the concepts. However, even with the use of strong biases, current machine learning techniques, and especially deep learning and reinforcement learning approaches, require a large number of examples [28].

Aware of this limitation, there has been a renewed interest in *teaching* computers [23, 40, 41], rather than just focusing on machine learning systems that can only expect examples at random. One of the key concepts in machine teaching is the power of choosing an optimal witness set [12, 32, 13, 15]. This set is chosen as small as possible, such that the learner still identifies the concept. However, for interesting, rich concept classes we do not know how to choose just a few examples that, in expectation, make an existing learning system find the solution. This contrasts strongly with the way humans teach other humans, where even very complex Turing-complete (universal) concept classes in natural language can be transmitted using just a few examples. For instance, when humans are told that "dollars", "euros" and "yens" are positive examples but "deutschemarks" are not, most understand that the concept is about currencies that are legal tender today. This kind of learning (or *teaching*, where the examples for the concepts are chosen, as with these words), is still an important challenge for AI. This is also related to natural language understanding, and the fact that humans often transmit concepts by example, rather than using the description of the concept. Teaching, either in machines or humans is a poorly-understood phenomenon that requires strong biases on sender and receiver, and the awareness from both peers that they are in a 'pedagogical situation' [31].

The teaching dimension of a concept [12, 32] in some concept class is the minimum number of examples required such that a learner uniquely identifies (learns) the concept, discarding all other concepts in the given concept class. The teaching dimension of a concept *class* is commonly understood as the worst case, which is usually unbounded for infinite concept classes. With the use of preferences (a kind of bias) we get some finite (worst-case) teaching dimensions for some restricted languages [14], but we suspect that these are unbounded for many other languages. The question is whether, for richer languages, we can still get finite, and even short, teaching dimensions *on average*? A uniform distribution, usually assumed for finite classes [2, 25], cannot be applied to infinite concept classes. The main insight comes if we realise that, apart from the *learning* preference or prior, we can consider a *sampling* prior, where not all concepts in the class have the same probability to be taught.

The learning prior makes the learner prefer some concepts over others, in the tradition of the K-dimension [3, 4] and the preference-based teaching dimension (PBTD) [14]. If the given witness set is consistent with (infinitely) many concepts, the one that is preferred will be output. But if we understand this complexity-based prior or preference as a probability, we can also see that as more examples are seen, we have increasing posterior probabilities. Actually, be it preferences, complexities or priors, it is not always easy to have a perfect identification of these preferences nor to have a perfect alignment between teacher and learner. Consequently, we need a teaching procedure that can reduce the uncertainty of a wrong identification. One interesting question is whether we can determine the minimum number of examples to get a given certainty. With very little notational effort we can generalise the K-dimension and PBTD to a situation where we define a teaching dimension given a certainty or probability level ρ. For instance, how many examples do we need so that the learner identifies the concept with probability at least 0.99? We will see how the teaching dimension can be extended under the notion of learning prior in order to answer this question.

The sampling prior, on the other hand, is used by the teacher (or tester) to see whether the learner is able to learn the whole class and

[1] Universitat Politècnica de València, Spain, email: jorallo@upv.es
[2] University of Bergen, Norway, email: Jan.Arne.Telle@uib.no

not just a particular subset of it. Consequently, it has to be as diverse (entropic) as possible. Note that the sampling prior is about a representative choice of concepts, not about the intentional choice of the examples for each concept.

Both priors are referring to how likely or expectable a concept is, and should be linked in some way. Indeed, we investigate whether this alignment between the learning prior ('chosen' by both learner and teacher) and the sampling prior (perhaps fixed or chosen by a tester) can lead to short example sets on average, ensuring that teaching sessions are feasible.

Of course, for every concept class one can always get a finite expected teaching dimension by putting almost all the mass of the distribution on a few concepts or choosing a sampling prior that decays fast enough. The question is whether, for some particular rich concept classes, there are some reasonable priors, still with infinite Shannon entropy [35], for which teaching is feasible. We observe, from the cases of Turing machines and finite state machines, that the more expressive the language is the more extreme (biased) the distributions must be in order to get teachability. But we will see that the distributions can still be sufficiently entropic at one end. This view creates a relation between the expressiveness of a language and how entropic the prior must be in order to make teaching possible, a more gradual alternative to the traditional (Chomskian) hierarchical view of languages. By fixing a probability level in the identification of the concept, we also link teaching to probabilistic inference, adding a certainty level to the trade-off.

In this paper, we analyse priors that are derived from complexity functions (program length, number of states, running times, etc.). This leads to the interpretation that if concept c_1 is simpler than c_2 then it will be given more probability by the learner given the same witness set, and it will be more likely to be sampled by the teacher. This also implies that if a learner has a prior, its representation language should be aligned with it, making more likely concepts require fewer resources in the language (as it happens with human language and, of course, in communication theory).

Given this new notion of expected teaching dimension, we obtain two major results. First, we get finite (and actually small) expected values for Turing-complete languages. This matches the observation of humans requiring very few examples when teaching or transmitting concepts in natural language. Second, we derive effective settings for a particularly interesting infinite concept class, the set of regular languages. In detail, we provide a series of contributions:

- We show that teaching for rich infinite concept classes can be done with a simplicity-based prior that is shared by learner and teacher (the learning prior). But this simplicity-based prior, when used for choosing the concepts (the sampling prior), still represents the whole concept class.
- We present a new conceptualisation of expected teaching dimension using the learning and the sampling prior. The learning prior is a probabilistic reformulation of the K-dimension and the Preference-Based TD (PBTD).
- We provide results showing that the expected teaching dimension for Universal Turing Machines (and hence other Turing-complete languages) is small, with the universal biases based on the program size of the concepts.
- Since universal biases based on Kolmogorov complexity are incomputable, we introduce computational time, using Kt for concept complexity. We get a computable learner but a teacher finding the smallest witness set is still non-computable.
- We show finite expected teaching dimension for regular languages using priors derived from the number of states of the minimal finite state machine (FSM) expressing the concept, proving both learner and teacher are computable.
- When the certainty in the identification is not considered, all the results –except when Kt is used– hold for the K-dimension and also for PBTD.
- When the prior is used to derive learner posteriors, we derive bounds for how many examples are needed to reach a given certainty of having identified the concept. This parametrises the teaching dimension taking it beyond the notion of preference to a degree of certainty in teaching.

TMs and FSMs are perhaps the two most important concept classes if we want to take machine teaching to really expressive and compositional scenarios. Parametrising by a probability level also enlarges the possibilities and flexibility of the teaching dimension.

2 Teaching posteriors: the learning prior

Let us first introduce the classical teaching dimension. We have a possibly infinite instance space X, with instances $x_i \in X$, that can be either positive examples, denoted by a pair $\langle x_i, 1\rangle$, usually represented as x_i^+, or negative examples, denoted by a pair $\langle x_i, 0\rangle$, usually represented as x_i^-. A concept is a binary function over X to the set $\{0, 1\}$. A concept language or class C is composed of a possibly infinite number of concepts. An example set S is just a (possibly empty) set of examples. We say that a concept c satisfies (or is consistent with) S, denoted by $c \vDash S$, if $c(x_i) = 1$ for the positive examples in S, and $c(x_i) = 0$ for the negative ones. All concepts satisfy the empty set. Given this, the teaching dimension (TD) of a concept c with respect to a class C can be defined as follows [12, 32]:

$$TD(c) \stackrel{\text{def}}{=} \min_S\{|S| : \{c\} = \{c' \in C : c' \vDash S\}\}$$

This minimal set is known as a witness set, and the teacher can assume that the learner will infer the concept given its witness set. Some further assumptions are needed. For instance, one can define "coding tricks" [3, 5], such as assuming a coding between instances and concepts, so that the j^{th} instance always corresponds to the j^{th} concept, so basically one only needs to send the "index" to identify the concept, as a lookup table. An appropriate way [16] to prevent this considers that whenever a learner identifies a concept c with an example set S, it must also identify c with any other superset of S that is also consistent with c (Goldman and Mathias's condition). The Recursive Teaching Dimension (RTD) [42, 9, 8] is a variant where concepts are taught with an order, starting for those of smallest dimension and removing the identified concepts for the following iteration. This becomes slightly more powerful than the classical teaching dimension but still compatible with Goldman and Mathias's condition. Additionally, RTD is related to the VC dimension, see e.g. [29, 8].

One thing to note about these settings is that extra examples (further confirming evidence) will not change the certainty of the learner about the concept. However, both machine teaching and learning are inductive processes where the reliability of a hypothesis can increase with confirming data by discarding alternative hypotheses. In other words, the classical teaching dimension is more about identification rather than inductive inference, and this holds also for the PBTD and K-dimension. These latter lower the witness size: in PBTD by a total order on concepts and requiring the learner to distinguish a concept only from concepts lower in the order, while the K-dimension is similar but uses a function from concepts to natural numbers instead of

a total order. However, we would like the learner to be increasing its confidence as it gets more examples, even past the identification.

We can reconcile this by considering that the learner has a prior, and as more examples are seen, more hypotheses are excluded, but at the same time the posterior of the remaining hypotheses is changing. So given a learning prior w on concepts of a class C, such that $\sum_c w(c) = 1$, we are going to define the posterior as follows. We first define a normalisation term as the overall a priori distribution mass of the consistent concepts so far, given a set S: $m_w(S) \stackrel{\text{def}}{=} \sum_{c \vDash S} w(c)$. The *teaching posterior* gives a probabilistic assessment for a concept c after seeing S, namely:

$$TP_w(c|S) \stackrel{\text{def}}{=} w(c|S) = \frac{w(c)}{m_w(S)} \text{ if } c \vDash S \text{ and 0 otherwise.} \quad (1)$$

Under this posterior, it is not only that Goldman and Mathias's condition is preserved but that the certainty of the identification usually increases as we add more elements to S. In other words, for all c if $S \subset S'$ then $w(c|S) \leq w(c|S')$ provided that $c \vDash S'$. There might even be cases where all competing hypotheses are excluded. In this case we have complete certainty that c is the intended concept.

Using this prior w, we define the teaching dimension as follows:

$$TD_w(c) \stackrel{\text{def}}{=} \min_S\{|S| : \{c\} = \arg\max_{c' \vDash S}\{w(c')\} \quad (2)$$

$$= \min_S\{|S| : \{c\} = \arg\max_{c'}\{TP_w(c'|S)\}\} \quad (3)$$

The expression on the top (2) is preferable when the prior is independent of the set chosen, while the one on the bottom (3) can accommodate cases where the prior (and hence the posterior) changes depending on the witness set, as we will see with Kt complexity.

Basically, for the teaching dimension without any uncertainty level or probability, the prior w introduces a preference when choosing among consistent hypotheses. In this case, it turns out to be an alternative formulation (quantitative, so necessarily a total order if concepts are arranged into batches of same w) to the preference-based teaching dimension (PBTD) [14], and ultimately more closely related to the K-dimension ([3, 4]), where this preference or ranking is linked to a measure of complexity, as we will revisit below. We also see explicitly that the classical teaching dimension is assuming that all concepts are equally likely (maximum entropy), which is unrealistic in many situations. For some infinite concept classes this would lead to the no-free-lunch theorems [38, 39]).

We introduce the parameterised version of the teaching dimension given a certainty or probability level ρ. In other words, the teaching dimension for confidence level ρ of a concept c is the size of the smallest set that uniquely identifies c while also assigning it a posterior probability greater than or equal to ρ:

$$TD_w^{[\rho]}(c) \stackrel{\text{def}}{=} \min_S\{|S| : \{c\} = \{c' : w(c'|S) \geq \rho\}\} \quad (4)$$

Let us see an example of how the priors are converted into posteriors, and how the posteriors increase as more concepts are discarded by the increase of the witness set, as in a truly inductive process. For the concept class in Table 1, when no example is given, $m_w(\emptyset) = 1$. The posteriors are still equal to the priors (e.g., the probability for c_4 is still 0.10). If x_4^- is presented, then we can discard c_2, c_3, c_6 and perhaps some other concepts in 'Rest'. Let us assume that half of the concepts in 'Rest' are discarded. This would lead to $m_w(\{x_4^-\}) = 0.30 + 0.10 + 0.06 + 0.015 = 0.475$ with the posterior probability for c_4 being now 0.10/0.475 = 0.21 (but not the highest of the compatible concepts, which is still c_1). If x_3^- is added to the set, then c_1 is now found inconsistent, and assuming that half of the remaining concepts in 'Rest' are discarded, we would have $m_w(\{x_4^-, x_3^-\}) = 0.10 + 0.06 + 0.0075 = 0.1675$ with the posterior probability for c_4 being updated to 0.10/0.1675 = 0.597. This is now the highest, which means that $TD_w(c_4)$ is not higher than 2, and since no single example can distinguish it from c_1, c_2, c_3, it is actually 2. Note that this concept c_4 can be suggested by the learner after seeing $\{x_4^-, x_3^-\}$ even if it is not the only compatible concept. Finally, if x_5^+ is shown, c_5 is now shown inconsistent and let us assume that this set discards half of the remaining in "Rest". Then $m_w(\{x_4^-, x_3^-, x_5^+\}) = 0.10 + 0.00375 = 0.10375$ and the posterior probability for c_4 will now be 0.10/0.10375 = 0.964. Thus we see that with TD, the posterior probabilities can still increase when receiving further consistent evidence.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	...	$w(c_i)$	TD	TD_w	$TD_w^{[.5]}$	$TD_w^{[.75]}$	$TD_w^{[.95]}$
c_1	0	0	1	0	1	1	0	...	0.30	∞	0	1	1	1
c_2	0	1	0	1	1	1	0	...	0.25	∞	1	1	1	2
c_3	1	0	0	1	1	1	0	...	0.20	∞	1	1	1	2
c_4	0	0	0	0	1	1	0	...	0.10	∞	2	2	3	3
c_5	0	0	0	0	0	1	0	...	0.06	∞	1	1	1	4
c_6	0	0	0	1	1	0	1	...	0.06	∞	1	1	1	4
Rest	-	-	-	-	-	-	-	...	0.03	.	.	.	.	.

Table 1. An infinite concept class with a learning prior w where the six most likely concepts only differ on seven examples. The 'Rest' row captures all other concepts. The teaching dimension varies with the confidence level.

Table 1 shows TD_w for no particular confidence (equal to setting $\rho = 0$) and then the teaching dimensions for different confidence values ρ (0.5, 0.75, 0.95). We see that the teaching dimension increases when we require higher confidence (posterior probability). Another interesting observation is that those concepts requireing fewer examples than other concepts for low confidence (e.g., $TD_w^{[.5]}(c_5) = 1 < TD_w^{[.5]}(c_4) = 2$), can require comparatively more than these other concepts when the confidence level grows (e.g., $TD_w^{[.95]}(c_5) = 4 > TD_w^{[.95]}(c_4) = 3$). This means that the ranking of concepts by TD changes with variable confidence ρ.

Of course, if we are only interested in identification, and not in quantifying certainty, this would be tantamount to the PBTD or K-dimension, and the actual numbers would not matter. In sections 4 and 5, we will only pay attention to the ranking of the concepts derived from the learning prior, and hence all the results[3] can be applied to the PBTD. However, in section 6 we will investigate the full possibilities of a probabilistic understanding of the learning prior. And now, let us pay attention to the sampling prior.

3 Expected teaching dimension: sampling prior

Up to this point, we have talked about the teaching dimension of one concept in a class. The teaching dimension of the whole class, and the classical worst-case scenario is defined as follows: $\max_{c \in C} TD_w(c)$. For many infinite concept classes, even with the use of a strong learning prior, there will not be an upper bound on the number of examples needed to distinguish the concepts. So, it becomes necessary to talk about an expected TD for a concept class C. This introduces a *sampling prior* v over concepts, which is used to obtain the expected TD for a concept class.

$$\mathbb{E}_v[TD_w(C)] \stackrel{\text{def}}{=} \sum_{c \in C} v(c) \cdot TD_w(c) \quad (5)$$

Of course, the result will strongly depend on the choice of v. One possible option is to assume $v(c) = w(c)$, meaning that the prob-

[3] Except for Kt, as the posterior not only depends on coverage and the prior, but also on the witness set.

ability that is used for calculating the plausibility of a concept (the learning prior) is the same as the probability of that concept to appear (the sampling prior). The key question comes with rich concept classes with infinitely many concepts and, as a result, infinitely many examples (otherwise some concepts would not be distinguishable by definition). We cannot choose a uniform distribution for neither w nor v if the class is infinite and discrete.

A natural idea when assigning a non-zero probability to an infinite discrete set of concepts is to use some distribution that is inversely related to the resources or complexity required by the concept, as given by a complexity function $K : C \to \mathbb{N}$ assigning a complexity value $K(c)$ for all concepts. This is actually the idea behind the K-dimension [3, 4]. However, we now need to apply this to the sampling distribution as well in order to calculate the expected teaching dimension. First, we assume that the learning prior is consistent with the complexity function, i.e., inversely monotonically related:

$$\forall c_1, c_2 \in C : w(c_1) \geq w(c_2) \Leftrightarrow K(c_1) \leq K(c_2) \tag{6}$$

From the infinitely many sampling distributions v, it makes sense to choose a distribution that is compatible with the learning distribution:

$$\forall c_1, c_2 \in C : v(c_1) \geq v(c_2) \Leftrightarrow w(c_1) \geq w(c_2) \tag{7}$$

which, from Eq. 6, implies that both distributions are monotonically related. Let us denote by C_k the "batch" composed of all the concepts of complexity k, i.e., $C_k = \{c : K(c) = k\}$. From Eq. 6, w and v are constant in each batch. The size of each batch is $N_k = |C_k|$. Then we add up all the sampling probabilities of the same batch, denoted by $V_k = \sum_{c \in C_k} v(c)$. The expected TD becomes:

$$\mathbb{E}_v[TD_w(C)] = \sum_{k=1}^{\infty} \frac{V_k}{N_k} \sum_{c \in C_k} TD_w(c)$$

The average TD_w for a batch k is given by $\frac{1}{N_k} \sum_{c \in C_k} TD_w(c)$. Consider an upper bound for this average, denoted by D_k. Then,

$$\mathbb{E}_v[TD_w(C)] \leq \sum_{k=1}^{\infty} V_k \cdot D_k \tag{8}$$

This means that once the batches are created by the complexity function, the expected TD only depends on the progression of the sampling distribution by batches and the progression of (a bound of) the average TD in the batch. Figure 1 show an example where the batched sampling distribution is geometric with parameter $1/6$, i.e., $V_k = (1/6) \cdot (5/6)^{k-1}$ with upper bound on average TD in the batch of $D_k = k^2$. With these parameters, the sum converges to a finite expected TD: 66. The geometric series k^2 is dominated by an exponential decay in V_k.

The relevant question is, once we achieve a bound D_k, can we think of a sampling distribution that can guarantee a bounded number of examples in the teaching sets *on average*? Even with the constraint given by Eq. 7, there are many distributions for v. One trivial case to minimise Eq. 8 is to choose v in such a way that it gives all the mass of the probability to one batch with low or minimal teaching dimension. Basically this would restrict the class to a finite distorted version. Consequently, a trade-off emerges between $\mathbb{E}_v[TD_w(C)]$ and v. More entropic (or diverse) sampling distributions v will be able to capture the whole of the concept class (and actually be *representative* of it) at the cost of having a higher expected TD. In any case, it is important to determine those distributions for which the expected TD is not bounded, because, for those, teaching will be impossible.

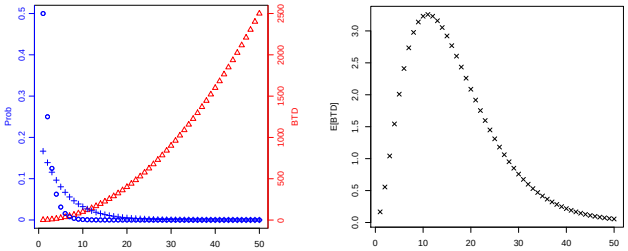

Figure 1. Left: The different components in the expected TD. The summed sampling prior V_k (blue crosses) for each batch k, and also the summed learning prior (blue circles). Also (red triangles) the (bound of the) average TD per batch k. Right: Components in the expected TD. The composition of the sampling prior with the TD gives the contribution of the expected TD for each value of k, whose sum in this case is finite (66).

It is then the relation between the teaching dimension using a learning prior and the sampling distribution used for expectation what we investigate next, for two very important concept classes: Turing machines and finite-state machines.

4 Expected TD for universal languages (TMs)

Turing machines represent the most general class for (traditional) computation. Consequently, the choices for w and v will connect with fundamental computational concepts such as Kolmogorov complexity, Solomonoff's prediction and inductive inference [27, 34]. For Turing machines, programs map to computable binary functions, as there are infinitely many for each concept. We say that a concept c is represented by program p in a universal Turing machine (UTM) M, denoted by $p \triangleright_M c$, if for every example $\langle x_i, b \rangle$ in c we get that the machine M, after being fed by the program p and an appropriate binary encoding of the example (examples are natural numbers) outputs the correct label, i.e., $M(\langle p, \sigma_i \rangle)$ writes b on the output string and halts. We now look for a measure of complexity of the concepts, so we extend the notion of Kolmogorov complexity as follows:

$$K_M(c) \stackrel{\text{def}}{=} \min_{p : p \triangleright_M c} \ell(p) \tag{9}$$

where $\ell(p)$ is the length of p in bits. In other words, the complexity of a concept is the length of the shortest program that represents (computes) the concept. We now define

$$U_M(c) \stackrel{\text{def}}{=} 2^{-K_M(c)} \tag{10}$$

which is a universal distribution over concepts based on their *algorithmic probability* [27]. To ensure that the sum is ≤ 1, M must be a prefix-free UTM. Still, since a concept can be represented by infinitely many programs[4], this U_M will not add up to one, but it can be normalised to make an actual distribution w. To highlight the dependency on the UTM chosen, we use notation TD_M when $w = U_M$. We now can simplify Eq. 2:

$$\begin{aligned} TD_M(c) &= \min_S\{|S| : \{c\} = \arg\max_{c' \vDash S}\{2^{-\min_{p : p \triangleright_M c'} \ell(p)}\}\} \\ &= \min_S\{|S| : \{c\} = \arg\min_{c' \vDash S, p : p \triangleright_M c'} \ell(p)\} \end{aligned}$$

[4] An alternative Epicurean formulation (where all the consistent programs are considered and not just one of them), more in the spirit of Solomonoff's algorithmic probability [34] would be $U'_M(c) \stackrel{\text{def}}{=} \sum_{p \triangleright c} 2^{-\ell(p)}$. Note that the difference between both formulations is tightly bounded, as if p is the shortest one, it will dominate this probability.

The last expression has a more natural interpretation and looks more similar to Balbach's complexity teaching dimension [3, 4], although we work with concepts that can be implemented by (infinitely) many programs each. From the above we see that what matters is the ranking, so all the results that follow will hold for PBTD too.

We now have to look at the sampling distribution v. A common choice here is yet again a universal distribution $v(c) = 2^{-K_M(c)}$. This means that for each concept whose shortest program has size k its probability is 2^{-k}. The probability of all the concepts in the batch is then $V_k = 2^{-k} \cdot N_k$. From here, we can instantiate Eq. 5 by batches as for Eq. 8:

$$\mathbb{E}_M[TD_M(C)] = \sum_{c \in C} 2^{-K_M(c)} \cdot TD_M(c) \leq \sum_{k=1}^{\infty} 2^{-k} \cdot N_k \cdot D_k \tag{11}$$

The question is how we can bound the average teaching dimension for batch k. From Kushilevitz et al. [24] we know that for a finite concept class C of binary vectors of length m we have that the average teaching dimension (assuming uniform bias $u(c) = 1/|C|$)), i.e., $\mathbb{E}[TD_u(C)]$, is bounded as follows:

$$\forall C : \mathbb{E}[TD_u(C)] \leq 2\sqrt{|C|}$$

Interestingly, for batch k, we only need to distinguish a concept from all the other concepts in its batch N_k, and the concepts in previous batches. Let us denote by $N_{\leq k}$ the number of concepts in batches 1 to k. This means that the average TD for C_k is bounded by $2\sqrt{N_{\leq k}}$.

But what is N_k?, i.e., how many concepts have shortest programs of size k? This cannot be 2^k, since it has to be a prefix coding. The actual value will depend not only on the UTM but also on the chosen coding. For instance, if we use a unary coding, we can get a convergent result very easily, since there is only one program for each k, so the term N_k would be 1 and the term $N_{\leq k}$ would be k. However, a unary coding is not universal.

We can try with Elias gamma coding [11, 30]. This is not asymptotically optimal, but it is stillack ofl universal. Basically, this coding uses a leading sequence of k zeros (which states the size of the string), followed by a 1 and then the traditional binary coding of a number. For instance, the first 10 codewords are 1, 010, 011, 00100, 00101, 00110, 00111, 0001000, 0001001, 0001010. As we can see, for each batch of the same size we have 2^i codewords with a size of $2i + 1$, with i being the index of batch starting at 0, and this gives an upper bound on N_k. So now we have:

Proposition 1. *The expected teaching dimension of concept class C assuming a universal distribution with an Elias gamma coding is finite, bounded by $1 + \sqrt{2}$.*

Proof. We have:

$$\mathbb{E}[TD_M(C)] = \sum_{k=1}^{\infty} 2^{-k} \cdot N_k \cdot 2\sqrt{N_{\leq k}}$$

As $k = 2i + 1$, $N_k = 2^{i-1}$, $N_{\leq k} = 2^{i+1} - 1$, we have:

$$\begin{aligned} \mathbb{E}[TD_M(C)] &\leq \sum_{i=0}^{\infty} 2^{-(2i+1)} \cdot 2^{i-1} \cdot 2\sqrt{2^{i+1} - 1} \\ &\leq \sum_{i=0}^{\infty} 2^{\frac{-i-1}{2}} = 1 + \sqrt{2} \end{aligned}$$

□

This means that with some universal codings we can have a finite expected TD. In other words, if a teacher samples concepts according to its universal distribution using an Elias gamma coding and both teacher and learner use the size of their programs as learning prior, then the number of examples needed to teach the concepts is finite in expectation. Of course, this is the case because the very small programs dominate the distribution. However, we can modify the UTM and the coding in such a way that a more uniform-like distribution happens for sizes k up to any arbitrary size k_s provided that from that point on the distribution decays as fast as above.

For the TD as defined above the learner and teacher are incomputable, since K is incomputable. Can we think of a similar computable procedure? For instance, given a language L, a concept class C and a concept c, the teacher should be able to *compute* the associated small teaching set S and the learner should *compute* c from it. To get a finite procedure we investigate the introduction of computational steps in the complexity function, inspired by Levin's Kt [26, 27]. We consider any finite example set S and define[5]:

$$Kt_M(S) \stackrel{\text{def}}{=} \min_{p : p \triangleright_M c \vDash S} \left\{ \ell(p) + \log \sum_{s \in S} \tau_M(p, s) \right\} \tag{12}$$

where $\tau_M(p, s)$ represents the runtime of executing program p on example s to get a result. Note that we have now defined Kt for example sets rather than for concepts, as we did for K. In this case, Kt does not create a prior or preference over the concepts, but over example sets[6]. This means that the teaching dimension is best seen in terms of the posterior, as per Eq. 3.

The original dovetail search of Levin's universal search is 2-dimensional on an increasing budget: over programs of increasing size and runtimes. Here, we add a third dimension: over increasing sizes of encodings of example sets. We get the following results:

Proposition 2. *Using Kt_M, for every M and c, if given a minimal teaching set S for c, a learner can by computable finite means identify the Kt-simplest program p such that program $p \triangleright_M c$ and $c \vDash S$.*

Proof. The learner will follow a dove-tailing approach with an increasing budget. With budget B on Kt_M the learner will enumerate all possible programs p' ensuring that $Kt \leq B$. Note that this enumeration and its execution is finite because of the τ term. For those programs inside the budget we discard those that are not consistent with the set. Once the enumeration for a budget is exhausted, the budget is increased by 1. Ultimately, the first program that accepts the examples in S in the budget will be found. The learner has identified the concept. □

So, if the teacher knew that the Kt-simplest program for a given set S is p with p being consistent with the concept that is to be taught, we would have a computable setting. However, this is problematic:

Proposition 3. *Using Kt_M, given an M and c, the generation of the minimal set S by the teacher is incomputable.*

Proof. The teacher can try a dovetail enumeration checking that all simplest programs are different from c, which is non-computable in

[5] All logarithms in this paper are binary.

[6] The definition over concepts would choose the empty set and would boil down to K. That's why we need S.

general. This can be seen by reduction from the undecidable predicate $Equiv(p_1, p_2)$, which tests equivalence of two TMs. We have two algorithms, Learner and Teacher. $Learner(S) = p$, where p is the simplest program compatible with all pairs in S. $Teacher(c) = S$, where S is the smallest set such that $Learner(S) = p'$, with $Equiv(c, p')$. We know $Learner$ is decidable. If $Teacher$ were decidable then we could decide $Equiv$ by $Equiv(p_1, p_2)$ if and only if $Learner(Teacher(p_1)) = Learner(Teacher(p_2))$. By contradiction, $Teacher$ is undecidable. □

Even if the teacher knows the shortest program p for a concept, there might be problems. For instance, if p cannot be identified for a budget, for the next budget new programs may appear that are compatible with the examples competing with it. These alternative programs can be more efficient than p (e.g., using partial look-up tables). This problem will appear for those programs whose time complexity increases exponentially (or even higher) in the size of the examples, and we may never find a witness set for p. There are possible solutions to be explored with bounded time or including the size of the proof to show that concepts are equal or not (so the class is reduced to Turing machines such that it can be proved or disproved equivalence to all simpler programs). We leave this as future work and focus on regular languages in the following section.

5 Expected TD for regular languages (FSMs)

Regular languages are defined by finite state machines (FSMs), a very well-known class of concepts in computer science. One of the advantages of using FSMs, over TMs, is that some of the ingredients needed for an effective (and computable) teaching setting are present for FSMs. We consider only deterministic FSMs, also called automata, or deterministic finite automata. First, there is an algorithm with time complexity $O(k \log k)$ to reduce any FSM on k states to an equivalent FSM on a minimum number of states [20], and secondly there is an algorithm linear in the number of states to test equivalence of two FSMs [21]. Two FSMs A and B are equivalent if their languages $L(A)$ and $L(B)$ are equal. As a concept is represented by its canonical FSM, the number of states k can be used as a natural complexity measure for regular languages.

So now we define our batches as in the previous section, using k for the number of states. We consider a binary alphabet. Now, the question is how to determine the two factors in Eq. 8.

For the term D_k, we use results of Dana Angluin [1] in the setting of 'identifying an unknown regular set from examples of its members and nonmembers'. In Angluin's setting a 'minimally adequate teacher' answers membership queries about the set and also gives counterexamples to wrong conjectures provided by the learner, with the latter being an example string in the symmetric difference between the correct set and the conjectured set, until the learner has identified the correct set. The collection of all positive and negative examples thus provided are gathered in an 'observation table'. Membership queries alone will not suffice to identify the language, for any finite number of examples there are infinitely many compatible regular languages. Note however, that equivalence queries alone are sufficient, the learner enumerates the regular languages non-decreasingly by number of states, and asks equivalence queries until arriving at the right language. Angluin's contribution is an efficient combination of membership queries and equivalence queries of the above form, with each counterexample increasing by one the minimum number of states needed for the language, until the correct minimum automaton is arrived at. Thus the examples in the final observation table will form a witness set that in our setting can be used by the learner, with no interaction, to identify an automaton with minimum number of states. The main result of Angluin is a learning algorithm L* to identify any regular language on k states in time polynomial in k and providing an upper bound on the size of the observation table.

Proposition 4. *[1] For any regular set U on k states the learner L* outputs a minimal automaton for U in time polynomial in k. The observation table has at most $(q+1)(k+m(k-1))k$ entries, where q is alphabet size, and m the maximum length of a counterexample, that can be bounded by $m \leq k$.*

Note that we will not be using the same interactive protocol as Angluin does, but as explained above the construction produces, at the end of the process, an observation table where each entry in the table can be used as an example in a witness set. This witness set will in our setting be sent directly from teacher to learner, with no interaction, and our learner (not L*) will be able to uniquely identify the correct language since the witness set distinguishes it from all other languages on at most k states. The result in Proposition 4 is a worst-case analysis, so for any language on k states there is a set of positive and negative examples in a table of this size that suffice to uniquely distinguish the language from all other regular languages on at most k states. Thus, we can conclude that for any regular language c on k states over an alphabet of size $q = 2$ we have $TD_w(c) \leq 3(k + k(k-1))k = 3k^3$.

And now we have to choose the sampling distribution V_k. Since we have shown that $TD_w(c) \leq 3k^3$, when there is an FSM for c with k states, we know that the average TD_w for the batch of k states is just given by $D_k = TD_w(c) \leq 3k^3$. In order to ensure convergence for the expected teaching dimension, we can choose the total sampling probability for the batch as $V_k = \alpha k^{-(4+\delta)}$, with $\delta > 0$. We just choose α to ensure that this Dirichlet series sums up to 1, i.e. $\sum_{k=1}^{\infty} V_k = 1$, which can be done e.g. by including a multiplicative factor. Since we know that $\sum_{k=1}^{\infty} k^{-(4+\delta)}$ is the Riemann zeta function $\zeta(4+\delta)$, then $\alpha = \zeta(4+\delta)^{-1}$. The actual v for each different FSM (and hence concept) is just defined as V_k/N_k. With this choice:

Proposition 5. *Choosing $V_k = \alpha k^{-(4+\delta)}$ with $\alpha = \zeta(4+\delta)^{-1}$, we get the following bound on average TD for regular languages when shorter minimal automata are preferred*

$$\mathbb{E}_v[TD_w(C)] \leq 3\frac{\zeta(1+\delta)}{\zeta(4+\delta)}$$

Proof. We use the sampling distribution V_k (note that we do not need to use $v = V_k/N_k$, as N_k is not necessary), and we get:

$$\mathbb{E}_v[TD_w(C)] \leq \sum_{k=1}^{\infty} V_k \cdot D_k$$

$$\leq \sum_{k=1}^{\infty} \alpha k^{-(4+\delta)} \cdot 3k^3 = \sum_{k=1}^{\infty} 3\alpha k^{-(1+\delta)} = 3\frac{\zeta(1+\delta)}{\zeta(4+\delta)}$$

Note that we can only express sums like $\sum k^{-x}$ with a Riemann zeta function when $x > 1$ (for 1 both diverge, but for $x < 1$, the series diverges but ζ gives negative values). As $\delta > 0$ we are allowed to do this in the above expression. □

A particular case when choosing $\delta = 1$ gives $\mathbb{E}_v[TD_w(C)] = 4.76$, as we have $\frac{\pi^2}{6} = 1.6449$ on the numerator and 1.0369 on the

denominator. For $\delta = 0.5$, we get $\mathbb{E}_v[TD_w(C)] = 7.43$. For large δ, the value cannot go below $\mathbb{E}_v[TD_w(C)] = 3$ as $\lim_{x\to\infty} \zeta(x) = 1$.

In order to get convergence we need $V_k = \alpha k^{-(4+\delta)}$, which decays fast, even for low values of δ. However, it does so only polynomially in k in contrast to the exponential decay for TMs. For Turing machines we got $V_k = 2^{-k} \cdot N_k$, which decays exponentially (note that k was the size of the program, and here V_k decays polynomially, but k is the number of states). Describing a FSM of k states requires a program that is exponential in k, based on the number of minimal such FSMs [10]. Actually, this highlights the transmission efficiency of our setting, as the following corollary shows:

Corollary 6. *With a learner using a learning prior w that decreases on the number of states, we have that in order to transmit a concept c for which there is a FSM with k states, a teacher would need at most $3k^3(2 + 2\lfloor log_2 k \rfloor)$ bits using Elias gamma coding.*

Proof. We know that $D_k = TD_w(c) \leq 3k^3$, and we saw that through our construction in Proposition 4 the size of the examples is $\leq k$. This means that for binary strings, we would need at most to code $3k^3$ strings with a number of bits of $2\lfloor log_2 k \rfloor + 1$ each, using Elias gamma coding. With an extra bit for coding the class of each example, we need at most $3k^3(2 + 2\lfloor log_2 k \rfloor)$ bits. □

6 Reducing teaching uncertainty

One of the motivations for introducing the learning prior was the derivation of a posterior quantifying the certainty of the identification by the learner, and increase it by larger (but hence non-optimal) witness sets. Given the two representational formalisms seen in the previous two sections, how do they extend in terms of the posteriors?

We first analyse the case of UTMs. Wa want to show that the certainty of identification can be quantified, so for sake of simplicity we will not aim for the best bounds, but note that with an efficient prefix coding, such as Elias coding, all bounds can be made tighter.

Lemma 7. *For every constant $k \geq 0$, if p is the shortest program for witness set S, then there is an S' (of at most size $|S'| \leq |S| + 2^{\ell(p)+k+1}$) such that all programs p' covering S' with $\ell(p') \leq \ell(p) + k$ are equivalent to p.*

Proof. If we take S, all the programs of size $< \ell(p)$ are excluded because p was the shortest program for S. About the programs p' where $\ell(p) \leq \ell(p') \leq \ell(p) + k$, for each of them that differs in behaviour from p we can find an example s to add to S. With a binary coding of programs, there are less than $2^{\ell(p)+k+1}$ programs p' where $\ell(p) \leq \ell(p') \leq \ell(p) + k$, and either they are equivalent to p or they must differ for at least one example. Adding all these examples s to S we build S', which makes the last expression $|S'| \leq |S| + 2^{\ell(p)+k+1}$. Note that if $k = 0$, we just consider the alternative $2^{\ell(p)}$ programs of the same size. □

Corollary 8. *Let us consider Kolmogorov complexity K as per Eq. 9 and a UTM M that gives a learning distribution for concepts $w(c)$ as per Eq. 10 with its posterior $w(c|S)$ defined as per Eq. 1. Then, for every computable concept c and certainty $0 \leq \rho < 1$ there is a finite S such that $w(c|S) \geq \rho$.*

Proof. We know that for every computable c there is an S and a program $p \triangleright_M c$ such that $Learner(S) = p$ using K. Consider p as the shortest one. We know that $w(c) = 2^{-\ell(p)}$. We also know that $m_w(S) = \sum_{c \vDash S} w(c) \geq 2^{-\ell(p)}$ but as no other equivalent program can be found of shorter size, we also know that $m_w(S) \leq 2^{-\ell(p)+1}$. So we have that $w(c|S) = w(c)/m_w(S) \geq 2^{-\ell(p)}/2^{-\ell(p)+1} = 0.5$. Now we can apply lemma 7 to increase this posterior. We can find an S' such that all programs p' covering S' with $\ell(p') \leq \ell(p)+k$ must be equivalent to p, so this means that all non-equivalent concepts c' covering S' must have complexity greater than $\ell(p) + k$, so:

$$
\begin{aligned}
m_w(S') &= \sum_{c' \vDash S'} w(c') \\
&= w(c) + \sum_{c' \vDash S', c' \neq c} w(c') \\
&= 2^{-\ell(p)} + \sum_{c' \vDash S', c' \neq c} w(c') \\
&= 2^{-\ell(p)} + \sum_{n \geq \ell(p)+k+1} \left\{ \sum_{c' \vDash S', c' \neq c, K(c')=n} w(c') \right\} \\
&= 2^{-\ell(p)} + \sum_{n \geq \ell(p)+k+1} \left\{ \sum_{c' \vDash S', c' \neq c, K(c')=n} 2^{-n} \right\} \\
&\leq 2^{-\ell(p)} + 2^{-\ell(p)-k} = 2^{-\ell(p)}(1 + 2^{-k})
\end{aligned}
$$

The last step derives from the fact that the probability mass cannot exceed 2^{-m} for all programs of size m, so that the mass of all programs of size $\geq m$ is $\sum_{\geq m} 2^{-m} = 2^{-m+1}$. Using $m = \ell(p)+k+1$ we get the last step. We then have

$$
\begin{aligned}
w(c|S') &= \frac{w(c)}{m_w(S')} \\
&\geq \frac{2^{-\ell(p)}}{2^{-\ell(p)}(1 + 2^{-k})} \\
&= \frac{1}{1 + 2^{-k}}
\end{aligned}
$$

and can get any value for ρ as close to 1 as we want by taking a large value of k. □

Recall that $l(p) + k + 1$ gives the lower limit on length of rival programs. The basic rationale is that the mass of rival programs is made smaller with higher k.

Proposition 9. *Given a concept c in a concept class C its teaching dimension with confidence ρ is bounded by $TD_M^{[\rho]}(c) \leq TD_M(c) + \frac{\rho}{1-\rho} 2^{\ell(p)}$, where p is the shortest program for c. The margin $k \leq -\log(1/\rho - 1)$ suffices for this.*

Proof. If the concept has $TD_M(c) = d$, then the shortest program p for c is identified with a witness set S of cardinality $|S| = d$. It is sufficient to apply lemma 7 for the right value of k to find a new S' with $|S'| \leq |S| + 2^{\ell(p)+k+1}$ to reach this ρ. From the proof of corollary 8, we can choose $\rho = \frac{1}{1+2^{-k}}$, so $k = -\log(\frac{1}{\rho} - 1)$, and

we plug this above and the size of S to get:

$$\begin{aligned} TD^{[\rho]}_M(c) &\leq d + 2^{\ell(p)-\log(\frac{1}{\rho}-1)} \\ &= d + \frac{\rho}{1-\rho}2^{\ell(p)} \\ &= TD_M(c) + \frac{\rho}{1-\rho}2^{\ell(p)} \end{aligned}$$

as the bound. □

Note that the bound on the teaching dimension grows exponentially as a bound on the size of the shortest program for the concept, meaning that increasing the certainty is more costly (in terms of teaching dimension) for concepts of high complexity (according to these bounds). The expression also brings insight to the situation when we use different UTMs. The invariance theorem [27] extends from programs to concepts as per Eq. 9. We see that the shortest program for a concept for UTM U cannot be larger than the shortest program for that concept for any other UTM V up to a constant that only depends on the two UTMs. This constant could be used to derive a bound in the teaching dimension when the machines differ.

In the case of UTMs, we explored the possibility of using Kt as a computable version of K. If we attempt to increase the teaching certainty, the definition we gave in Eq. 12 sums the runtime of all the examples in the witness set. This means that making the set larger will decrease the posterior, so finding bounds for the teaching dimension using Kt becomes more convoluted. An option for future work would be to redefine the posterior, so it becomes computable and not growing with the cardinality of the witness set (e.g., average runtime rather than the sum).

Finally, for FSMs, we believe that the results in this section can be extended with some of the priors we used in section 5, getting bounds on the increase of the teaching dimension to get more confidence.

In general, estimating the posterior, or bounding the increase of the teaching dimension given the desired confidence, is useful to give more stability to teaching and making it less dependent on misalignments between teacher and learner. As in the K-dimension and the PBTD, both teacher and learner have to share exactly the same complexity function or preference function. With the use of a confidence margin, one can admit some bounded discrepancies in the prior.

7 Discussion

Analysing whether and how infinite concept classes can be taught led us to a dilemma between making the teaching set finite on average and the use of a wide, entropic sampling distribution actually covering the whole class. The observation that humans are able to cover a wide range of concepts and can learn from very few examples suggests that humans share a prior and may communicate, and teach, accordingly. This strong bias may well depend on the application, domain or context, but it is natural to make it related to (or based on) the complexity of the concept, as we have investigated here, very much in the same way to other theories of inductive inference such as Solomonoff's prediction, the use of Occam's razor, structural risk minimisation or the MML/MDL principles [34, 37, 27]. Therefore, we can think of this work as bringing the above setting from the standard learning scenario to the teaching scenario, with further connections to be unveiled with possibly more positive results. In practice, these ideas have worked well for learning from very few examples in areas such as inductive programming, programming by examples or teaching by demonstration [18, 17, 19, 33], usually without recognising the two different priors involved.

The notion of simplicity for TMs depends on the choice of the UTM. Similarly, for FSMs, the number of states is a natural measure of simplicity, but others could be used, such as the length of the shortest regular expression expressing the concept. The invariance theorem [27] establishes that simplicity is the same up to a constant that is independent of the concept, but this constant can be large. This motivates a possible study of other versions of the TD, more independent from the particular complexity measure. In this paper, the version that takes a confidence level leads to a new trade-off between certainty and teaching dimension, which only affects the robustness of the identification when the languages (complexities or preferences) of the teacher and learner are slightly different.

Another interesting thing to analyse is to consider the complexity function as a measure of difficulty of the concept and consider the session as an evaluation process. In this case, the sampling distribution could be adapted in such a way that, if we know the ability of the learner, we could sample concepts of appropriate complexity. In other words, the sample distribution could assign very low probability to the very easy concepts (small complexity) but still (necessary) decreasing probability from some given complexity, resembling a Poisson distribution, and breaking the monotonicity of Eq. 7.

The perspective from evaluation also helps us understand that the sample prior is not chosen to get a finite teaching dimension on expectation, but a way of modelling that all concepts from the class are not equally likely. This may be better understood by distinguishing a third actor, the evaluator, who chooses the concepts for the teacher. The evaluator has a syllabus, or a book, in which some concepts are more relevant, and hence likely, than others. In rich languages, simply by resource constraints and the difficulty of working with very complex concepts, it is natural to assume that the sample distribution will be strongly decreasing on program size, which is what the evaluator should focus on.

Similarly, the role of the learning prior has to be well understood. Unlike the sampling prior, in our setting (and the traditional machine teaching setting), the teacher chooses an optimal set according to the learning prior. The teacher does not sample from the learning prior to get those examples that would make the learner identify the concept with higher probability. This stochastic setting of the teacher would lead to suboptimal witness sets, but may be more realistic in human teaching. This is exactly the configuration that Shafto et al. [31] explore, using a Bayesian approach. In our case, the confidence is seen in terms of confirmatory evidence over the alternative hypotheses, and not in terms of identification, as in early learning settings [7]. Note that our setting is not interactive or incremental, and the order of the examples is irrelevant (unlike [6]), as the learner runs the algorithm over the whole set.

The analysis of complex concept classes is sometimes avoided for the batch setting in machine teaching because positive results are elusive. But there is a long tradition in machine learning and machine teaching where some positive results have been found for other formulations. For instance, for regular languages, in an interactive teacher-learning scenario, if the learner can send the hypothesis and the teacher replies with the lexicographically-first example that contradicts the hypothesis (if it is incorrect) then Ibarra et al [22] show that learning can happen in polynomial time. This and other settings are quite far from our scenarios where examples come as a set, but we leave it as future work to explore the connections, and the expected teaching dimension in particular, with these approaches. Another deviation from the traditional setting in machine teaching is by considering the size of the examples in the witness set, and not only their number. This different setting is considered in [36], but sticking

to the minimum certainty of identification (no ρ), unlike we do here.

In the common setting for machine teaching that we use in this paper, the very notion of expected teaching dimension forces us to consider non-uniform distributions. This work has made clear that a trade-off is necessary between an effective teaching and a wide coverage of the concept class. This gives several insights about how biases have to be embedded and used by learner and teacher, and also suggestions about efficient concept understanding and communication in general.

Acknowledgments

We thank David Dowe for some comments on an early version of this paper. We also thank Damián López for his insight and help about the size of the counterexamples that we consider when adapting [1].

REFERENCES

[1] Dana Angluin, 'Queries and concept learning', *Machine Learning*, **2**, 319, (1987).

[2] Martin Anthony, Graham Brightwell, and John Shawe-Taylor, 'On specifying boolean functions by labelled examples', *Discrete Applied Mathematics*, **61**(1), 1–25, (1995).

[3] Frank J. Balbach, *Models for algorithmic teaching.*, Ph.D. dissertation, University of Lübeck, 2007.

[4] Frank J Balbach, 'Measuring teachability using variants of the teaching dimension', *Theoretical Computer Science*, **397**(1-3), 94–113, (2008).

[5] Frank J Balbach and Thomas Zeugmann, 'Recent developments in algorithmic teaching', in *Intl Conf on Language and Automata Theory and Applications*, pp. 1–18. Springer, (2009).

[6] Frank J Balbach and Thomas Zeugmann, 'Teaching randomized learners with feedback', *Information and Computation*, **209**(3), 296–319, (2011).

[7] Janis Barzdins, Rusins Freivalds, and Carl H Smith, 'Learning with confidence', *Lecture Notes in Computer Science*, **1046**, 207–218, (1996).

[8] Xi Chen, Yu Cheng, and Bo Tang, 'On the recursive teaching dimension of VC classes', in *NIPS*, 2164–2171, Curran, (2016).

[9] Thorsten Doliwa, Gaojian Fan, Hans Ulrich Simon, and Sandra Zilles, 'Recursive teaching dimension, vc-dimension and sample compression', *Journal of Machine Learning Research*, **15**(1), 3107–3131, (2014).

[10] Michael Domaratzki, Derek Kisman, and Jeffrey Shallit, 'On the number of distinct languages accepted by finite automata with n states', *Journal of Automata, Languages and Combinatorics*, **7**(4), 469–486, (2002).

[11] Peter Elias, 'Universal codeword sets and representations of the integers', *IEEE transactions on information theory*, **21**(2), 194–203, (1975).

[12] Rūsinš Freivalds, Efim B Kinber, and Rolf Wiehagen, 'Inductive inference from good examples', in *International Workshop on Analogical and Inductive Inference*, pp. 1–17. Springer, (1989).

[13] Rusins Freivalds, Efim B. Kinber, and Rolf Wiehagen, 'On the power of inductive inference from good examples', *Theoretical Computer Science*, **110**(1), 131–144, (1993).

[14] Ziyuan Gao, Christoph Ries, Hans Ulrich Simon, and Sandra Zilles, 'Preference-based teaching', *Journal of Machine Learning Research*, **18**, 31:1–31:32, (2017).

[15] Sally A Goldman and Michael J Kearns, 'On the complexity of teaching', *J. of Computer and System Sciences*, **50**(1), 20–31, (1995).

[16] Sally A Goldman and H David Mathias, 'Teaching a smart learner', in *Conf. on Computational learning theory*, pp. 67–76, (1993).

[17] Sumit Gulwani, 'Programming by examples: Applications, algorithms, and ambiguity resolution', in *Intl Joint Conf on Automated Reasoning*, pp. 9–14. Springer, (2016).

[18] Sumit Gulwani, José Hernández-Orallo, Emanuel Kitzelmann, Stephen H Muggleton, Ute Schmid, and Benjamin Zorn, 'Inductive programming meets the real world', *Comm. of the ACM*, **58**(11), (2015).

[19] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil, 'Showing versus doing: Teaching by demonstration', in *NIPS*, 3027–3035, Curran, (2016).

[20] John Hopcroft, 'An $n \log n$ algorithm for minimizing states in a finite automaton', *Theory of machines and computations*, 189–196, (1971).

[21] John Hopcroft and Richard Karp, 'A linear algorithm for testing equivalence of finite automata', Technical Report 0, Dept. of Computer Science, Cornell U, (December 1971).

[22] Oscar H. Ibarra and Tao Jiang, 'Learning regular languages from counterexamples', *J. Comput. Syst. Sci.*, **43**(2), 299–316, (1991).

[23] Faisal Khan, Bilge Mutlu, and Xiaojin Zhu, 'How do humans teach: On curriculum learning and teaching dimension', in *Advances in Neural Information Processing Systems*, pp. 1449–1457, (2011).

[24] Eyal Kushilevitz, Nathan Linial, Yuri Rabinovich, and Michael Saks, 'Witness sets for families of binary vectors', *Journal of Combinatorial Theory, Series A*, **73**(2), 376–380, (1996).

[25] Homin K Lee, Rocco A Servedio, and Andrew Wan, 'DNF are teachable in the average case', *Machine Learning*, **69**(2-3), 79–96, (2007).

[26] Leonid A. Levin, 'Universal Search Problems', *Problems Inform. Transmission*, **9**, 265–266, (1973).

[27] Ming Li and Paul Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, 3rd Ed. Springer, 2008.

[28] Gary Marcus, 'Deep learning: A critical appraisal', *arXiv preprint arXiv:1801.00631*, (2018).

[29] Shay Moran, Amir Shpilka, Avi Wigderson, and Amir Yehudayoff, 'Compressing and teaching for low vc-dimension', in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, ed., Venkatesan Guruswami, pp. 40–51. IEEE Computer Society, (2015).

[30] Khalid Sayood, *Lossless compression handbook*, Academic Press, 2002.

[31] Patrick Shafto, Noah D. Goodman, and Thomas L. Griffiths, 'A rational account of pedagogical reasoning: Teaching by, and learning from, examples', *Cognitive Psychology*, **71**, 55 – 89, (2014).

[32] Ayumi Shinohara and Satoru Miyano, 'Teachability in computational learning', *New Generation Computing*, **8**(4), 337–347, (1991).

[33] Chengxun Shu and Hongyu Zhang, 'Neural programming by example', in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, eds., Satinder P. Singh and Shaul Markovitch, pp. 1539–1545. AAAI Press, (2017).

[34] R. J. Solomonoff, 'A formal theory of inductive inference. Part I', *Information and control*, **7**(1), 1–22, (1964).

[35] Kohtaro Tadaki, 'The Tsallis entropy and the Shannon entropy of a universal probability', in *2008 IEEE International Symposium on Information Theory*, pp. 2111–2115, (July 2008).

[36] Jan Arne Telle, José Hernández-Orallo, and Cèsar Ferri, 'The teaching size: computable teachers and learners for universal languages', *Machine Learning*, **108**(8-9), 1653–1675, (2019).

[37] C. S. Wallace and D. M. Boulton, 'An information measure for classification', *Computer Journal*, **11**(2), 185–194, (1968).

[38] David H Wolpert, 'The lack of a priori distinctions between learning algorithms', *Neural computation*, **8**(7), 1341–1390, (1996).

[39] David H Wolpert and William G Macready, 'No free lunch theorems for optimization', *IEEE Trans on evolutionary computation*, **1**(1), 67–82, (1997).

[40] Xiaojin Zhu, 'Machine teaching for Bayesian learners in the exponential family', in *Neural Information Processing Systems 26*, 1905–1913, Curran, (2013).

[41] Xiaojin Zhu, 'Machine teaching: An inverse problem to machine learning and an approach toward optimal education.', in *AAAI*, pp. 4083–4087, (2015).

[42] Sandra Zilles, Steffen Lange, Robert Holte, and Martin Zinkevich, 'Models of cooperative teaching and learning', *Journal of Machine Learning Research*, **12**(Feb), 349–384, (2011).