Teaching with Limited Information on the Learner's Behaviour

Ferdinando Cicalese
University of Verona, Italy
ferdinando.cicalese@univr.it

Sergio Filho
PUC-Rio, Brazil
sfilhofreitas@gmail.com

Eduardo Laber
PUC-Rio, Brazil
laber@inf.puc-rio.br

Marco Molinaro
PUC-Rio, Brazil
molinaro.marco@gmail.com

Abstract

Machine Teaching studies how efficiently a Teacher can guide a Learner to a target hypothesis. We focus on the model of Machine Teaching with a black box learner introduced in [Dasgupta et al., ICML 2019], where the teaching is done interactively without having any knowledge of the Learner's algorithm and class of hypotheses, apart from the fact that it contains the target hypothesis h^*. We first refine some existing results for this model and, then, we study new variants of it. Motivated by the realistic possibility that h^* is not available to the learner, we consider the case where the teacher can only aim at having the learner converge to a best available approximation of h^*. We also consider weaker black box learners, where, in each round, the choice of the consistent hypothesis returned to the Teacher is not adversarial, and in particular, we show that better provable bounds can be obtained for a type of Learner that moves to the next hypothesis smoothly, preferring hypotheses that are close to the current one; and for another type of Learner that can provide to the Teacher hypotheses chosen at random among those consistent with the examples received so far. Finally, we present an empirical evaluation of our basic interactive teacher on real datasets.

1 Introduction

Machine Teaching studies how efficiently a Teacher can teach a target hypothesis to a Learner. The classic works [22, 11] consider the setting where the Teacher sends in one shot a set of labeled examples to the Learner, which then has to output the correct target hypothesis. In more recent works, the focus has been on the interactive setting [15, 6, 16, 7] — where the Teacher and Leaner interact over multiple rounds. In each round, the Teacher sends examples to the Learner, which returns [1] some feedback; this process continues until the Learner reaches the target hypothesis (or a good approximation of it).

Machine teaching models have proved useful in several contexts, e.g., crowd sourcing [12, 24], intelligent tutoring systems [20, 25], analysis of training set attacks [18]. Moreover, commercial tools are under development by the Microsoft Machine Teaching Group, as detailed on their web page, which are based on, or employ, the paradigm of machine teaching, e.g., PICL, which leverages the selection of examples that maximize the training value of the interaction with the teacher; LUIS for natural language understanding; and other projects on building models for autonomous systems.

Most of the above works assume that the Teacher has significant knowledge about the Learner, e.g., its hypothesis class and the specific procedure employed for learning a hypothesis

[1] Unless specified, we will tacitly assume that Learner and Teacher are machines, hence we use neutral pronouns.

from labeled examples. However, this assumption excludes many important situations as human teaching and automatic learners with black box behaviour (e.g. Deep Nets). Thus, recent work in the field has focused on analysing scenarios in which the Teacher's knowledge about the Learner is limited [16, 7].

In particular, [7] addressed machine teaching with a *black box* learner, where the only knowledge of the Teacher about the Learner is that its hypothesis class contains the target. They considered a model of interaction where at each round the Teacher sends labelled examples to the Learner and it provides a hypothesis that is consistent with all the examples received so far. The authors provide bounds on the number of examples required to teach the target hypothesis to *worst-case* learners.

Here, we refine some existing results for the model of machine teaching with a black box learners considered in [7] and also introduce and analyse new variants of it. We are motivated, on the one hand, by the realistic scenario in which 'exact teaching' is not possible (the target does not belong to Learner's hypothesis class) and, on the other hand, by the fact that the Learner may not be adversarial to the Teacher.

1.1 Notation and Model

Before precisely stating our contributions we set some notation and explain the teaching model in more detail. There is a set $\mathcal{X}$ of examples, and a finite set $\mathcal{Y}$ of possible labels for each example. By a hypothesis we mean a function that maps each example in $\mathcal{X}$ to a label in $\mathcal{Y}$. We assume that:

Teacher: has a target hypothesis $h^* : \mathcal{X} \to \mathcal{Y}$, unknown to Learner ($h^*(e)$ is the correct label for e).

Learner: has a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, unknown to Teacher

In each round the Teacher sends the Learner a set of labeled examples $(e, h^*(e))$, then the Learner returns a hypothesis from its class with minimum number of errors in the examples received thus far. The goal of the Teacher is to send a minimum number of examples so as to make the Learner return a hypothesis from $\mathcal{H}$ with the smallest total number of errors in the whole dataset $\mathcal{X}$ (in the realizable case $h^* \in \mathcal{H}$, this means returning the correct hypothesis h^*). In the basic setting the Teacher does not have additional information on the learning algorithm used by the Learner to select among its minimum error hypotheses.

A fundamental notion in machine teaching is that of a *teaching set* for h^* [11], which is a set of examples $X \subseteq \mathcal{X}$ that distinguishes h^* from every other hypothesis in $\mathcal{H}$, that is, for every $h \neq h^*$ there is an example $e \in X$ for which $h(e) \neq h^*(e)$. We use $\mathcal{TS}(\mathcal{H}, h^*)$ to denote the size of the smallest teaching set for h^*. When $(\mathcal{H}, h^*)$ is clear from the context we use $\mathcal{TS}$ as a shorthand. We use $m = |\mathcal{X}|$ and $n = |\mathcal{H}|$ to denote the size of the sets of examples and hypotheses, respectively, and use $wrong(h)$ to denote the set of examples in which h fails (differs from h^*).

1.2 Contributions and Related Work

Our first contribution is a teaching algorithm $\mathcal{A}_{\text{base}}$ that *with high probability* guarantees convergence to the target hypothesis h^* using $O(\mathcal{TS} \log m \log n)$ examples (Theorems 1 and 2). Its correctness relies on a novel analysis of the on-line set cover algorithm proposed by [1]. The main obstacle to derive this analysis is handling the non-trivial dependence between the Teacher's and the Learner's actions over time. We rely on martingale techniques and arguments reminiscent of decoupling [8] to overcome this difficulty.

[7] present interesting results for black box learners and one of them is also a teaching algorithm based on a (different) adaptation of the on line set cover algorithm from [1]. Their

analysis guarantees bounds similar to ours, although it is based on the knowledge of an upper bound on n. However, some relevant subtleties arising from the interdependence between Teacher and Learner were not addressed in the proofs from [7]. In this respect, we understand that an additional contribution of our analysis is to clarify and formalize (via an application of our Lemma 2) the validity of a key statement in their argument (Lemma 5 of [7]). Details are presented in Suppl. Material, Appendix A. That said, we would like to emphasize that the algorithm and the statements from [7] are correct.

We also use our algorithm $\mathcal{A}_{\text{base}}$ as a basis for both improved results and extension to other variants of the problem. In Section 2.2, we propose a modified algorithm that obtains a stronger bound that depends on the (unknown) distribution of the number of errors among the hypotheses in $\mathcal{H}$ (Theorem 3). In Section 2.3, we generalize the above bound to the *non-realizable case*, where the Teacher cannot assume that the Learner's class contains the target hypothesis. Our algorithm guarantees that the Learner converges to the hypothesis $\tilde{h}$ that is the closest to h^* in its hypothesis class, after receiving $O(\mathcal{TS}_k \log m \log(m+n))$ examples, where $\mathcal{TS}_k$ is a lower bound on the number of examples needed for this task.

These results are valid for the *worst-case* Learner model [22, 11], where no assumption is made on which hypothesis the Learner selects among those of minimum error in the examples received thus far. Different models for the Learner's behavior have been recently considered [26, 10, 6, 17, 13]. The assumption that the Learner smoothly navigates over its hypothesis class, always updating its current hypothesis to one that is 'close' to it, was used to motivate the *local preference* model introduced in [6] and extended in [17]. For this type of Learner, when the closeness is measured in terms of Hamming distance, we present a teaching algorithm that with high probability sends $O(\mathcal{TS} \log n(\log err_1 + \log\log n))$ examples, where err_1 is the number of errors of the first hypothesis provided by the Learner (Theorems 5 and 6)[2] . Thus, this teaching algorithm benefits from a Learner that starts close to the target hypothesis. It is possible to show that this bound is not achievable by efficient algorithms, in the worst-case model, through a simple modification of a lower bound presented in [14].

We also obtain improved bounds for the model where the Teacher can ask the Learner for a batch of random hypotheses consistent with the examples presented thus far. We propose a teaching algorithm that, with high probability, teaches h^* by sending in total $O(\mathcal{TS} \log(n+m))$ examples and requesting $O(\mathcal{TS} \log(m+n))$ random hypotheses per round (Theorem 7). For the relevant case where the number of hypotheses n is larger than the number of examples m, the bound on the number of examples is the best possible for poly-time algorithms under the assumption $\mathcal{P} \neq \mathcal{NP}$, even when the Teacher knows the class of hypotheses $\mathcal{H}$ [21].

We note that our non-adversarial models for learners are related to models that have already been considered [6, 4, 23, 2]. In fact, the model of smooth transitions can be seen as an instance of the local preference model from [6], where hypotheses close to the current one, in terms of the Hamming distance, are preferred. Moreover, learners that return a random hypothesis have been considered in [4, 23]. However, in these works, in contrast to ours, the Teacher is aware of the Learner's hypothesis class. An analogous result, in a different context of teaching, where separation is proved between worst case and random adversary is [2].

Although the teaching algorithms mentioned so far aim at obtaining a small teaching set, they may end up producing a non-minimal one (w.r.t. example deletion). In Section 4 we show that with a limited amount of extra interactions the Teacher is able to construct a minimal teaching set. This result may be useful when the main goal is to obtain a compressed training set.

[2] We remark that this result holds if $n = |\mathcal{H}|$ is redefined as the number of non-equivalent hypotheses in $\mathcal{H}$ w.r.t. h^*, where two hypotheses are equivalent if they agree with h^* in exactly the same examples of $\mathcal{X}$; hence, $n \leq 2^m$ and $\log\log n \leq \log m$.

Finally, to complement our theoretical results, we present in Section 5 experiments with 12 real datasets that show that our basic Teacher (the $\mathcal{A}_{\text{agno}}$ from Section 2) sends significantly fewer examples, to reach a given level of accuracy, than a Teacher that does not interact with the Learner.

2 Teaching with Worst-case Learner

Recall the teaching model from Section 1.1. In this section we consider worst-case learners that can return any hypothesis $h \in \mathcal{H}$ that has smallest number of errors on the examples received thus far.

2.1 Realizable Hypothesis Case

We first consider the realizable case when the target hypothesis h^* belongs to the learner's class $\mathcal{H}$. Notice that in this case the Learner always sends a hypothesis that is correct on all examples seen thus far.

As in [7], we leverage the connection between teaching and *set cover*. We say that an example $e \in \mathcal{X}$ *covers* hypothesis $h \in \mathcal{H}$ if the latter makes a mistake in this example, namely $h(e) \neq h^*(e)$. Notice that covered hypotheses are out of consideration from the Learner, namely it never sends a hypothesis that is covered by the examples it has received. Thus, after the examples sent by the Teacher cover all hypotheses other than h^* the Learner must send back the correct hypothesis h^*, achieving the learning goal. This means that one can reduce the problem of teaching to that of *online set cover*: in the beginning of each round, the Teacher receives a hypothesis from Learner and sends examples that cover this hypothesis (and hopefully other unknown hypotheses), in a way that the total number of examples sent is small.

Our proposed teaching algorithm $\mathcal{A}_{\text{base}}$ uses the online set cover algorithm of [1], and can be described as follows (see Figure 1). It maintains weights W_e^t over the examples $e \in \mathcal{X}$ for each round t. When a new hypothesis h comes from the Learner (so it is not covered by the examples thus far), the Teacher first verifies whether $h = h^*$. If so, it accepts h. Otherwise, it increases in exponential fashion the weights of the examples where h is wrong until the sum of these weights becomes at least 1; then it randomly sends examples to Learner with probability proportional to the increase of the weights of the examples in this round. If h is neither accepted nor covered (i.e., no example is sent) by the end of the round, the algorithm returns FAIL.

While our algorithm is based on [1] the main novelty is in its analysis: the hypotheses ("elements" to be covered) depend on the examples ("sets") sent, and not only the analysis in [1] does not allow such dependencies but it also known that m examples are required for more general dependencies [[14], Theorem 2.1.3]. However, the dependencies that arise in this context of teaching are just so that we can handle them using martingale techniques.

Theorem 1. *Consider teaching a worst-case learner. In the realizable case $h^* \in \mathcal{H}$, algorithm $\mathcal{A}_{base}$ (in Fig. 1) initialized with $N \geq n$ and $\omega = m$ always sends $O(\mathcal{TS} \log N \log m)$ examples, and returns the correct hypothesis h^* with probability at least $1 - \frac{1}{N}$.*

Proof of Theorem 1. The first important observation is that the algorithm terminates in at most $O(\mathcal{TS} \log \omega)$ rounds [[1], Lemma 1]. In addition, since in each round the Teacher sends at most $O(\log N)$ examples we get the following.

Lemma 1. *$\mathcal{A}_{base}$ sends $O(\mathcal{TS} \log \omega \log N)$ examples.*

```
Algorithm $\mathcal{A}_{\text{base}}$
Input: Examples $\mathcal{X}$, (guess of) the number of Learner's hypotheses $N$, initial weight_ parameter $\omega \geq 0$
  1. Initialize weights $W_e^0 = \frac{1}{2\omega}$ for all examples $e \in \mathcal{X}$
  2. For each round $t = 1, 2, \ldots$:
     • Receive hypothesis $H_t \in \mathcal{H}$ from the Learner
     • If $H_t$ is correct in all examples (i.e., $H_t = h^*$), stop and return $H_t$
     • (Weight update) Double the weights of all wrong examples until their weight adds
       up to at least 1. That is, define
         $W_e^t = \begin{cases} 2^\ell \cdot W_e^{t-1} & \text{, if } e \in \text{wrong}(H_t) \\ W_e^{t-1} & \text{, if } e \notin \text{wrong}(H_t), \end{cases}$
       where $\ell$ is the smallest non-negative integer such that $W^t(H_t) := \sum_{e \in wrong(H_t)} W_e^t \geq 1$
     • (Sending examples) For every example $e$, let $D_e^t := W_e^t - W_e^{t-1}$ be the weight increase
       of example $e$ (note $D_e^t = 0$ if $H_t$ is not wrong on $e$)
       Repeat $4 \log N$ times: sample at most one example so that $e$ is sampled with probability
       $D_e^t$, and send it to Learner together with its correct label (note that $H_t$ is wrong on this
       example)
     • If no examples were sent, return FAIL
```

Figure 1: Teacher's algorithm for teaching a realizable hypothesis

So we only need to upper bound the probability that the algorithm returns FAIL. Let $W^t(h) := \sum_{e \in \text{wrong}(h)} W_e^t$ be the weight of h at the end of round t, and let $D^t(h) := W^t(h) - W^{t-1}(h)$ be the increase of this weight at round t. The intuition why the failure probability should be low is the following: If the algorithm fails on a hypothesis h it means that its weight is at least 1 by the end of the interaction and no example that covers h was sent. Let X_t be an indicator variable that is equal to 0 if no examples that cover h are sent at round t. We have $Pr[X_t = 0] = (1 - D^t(h))^{4 \log N}$, so the failure probability *should be* about $\prod_t (1 - D^t(h))^{4 \log N} \approx e^{-(4 \log N) \sum_t D^t(h)} \leq e^{-2 \log N} = 1/N^2$, where the last inequality holds because, at the beginning, the weight of h is at most $1/2$ and by the end of the algorithm is at least 1. By taking the union bound over all hypotheses $h \in \mathcal{H}$ we would conclude that the failure probability is at most $1/N$.

The problem is that this argument ignores crucial stochastic dependencies: the actual examples sent affect (through the Learner's response) the evolution of the weight of h, so that the set of random variables $\{X_t\}$ are not independent and, hence, we cannot take the product of probabilities as above. To handle this situation we abstract it as a sequence of dependent Bernoulli random variables X_t's whose biases (corresponding to $1 - (1 - D^t(h))^{4 \log n}$) depend on the history. Our main technical lemma shows that, regardless of the correlations, the probability that none of the indicators X_t is active is what we expect.

Lemma 2 (Adaptive Bernoullis). *Consider a finite probability space with filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots$ and let $X^1, \ldots, X^n \in \{0, 1\}$ be an adapted sequence of Bernoulli random variables, possibly correlated. Let $Z^t := \Pr(X^t = 0 \mid \mathcal{F}_{t-1})$ be the conditional probability that X_t is 0. Then for any stopping time τ w.r.t. $\mathcal{F}$ and $\alpha \geq 0$*

$$\Pr\left(X^1 = \ldots = X^\tau = 0 \quad \textit{and} \quad \prod_{t \leq \tau} Z^t \leq \alpha \right) \leq \alpha.$$

Proof. We can assume without loss of generality that there is no stopping time (i.e., τ always equals n): we can apply the result to the variables $\tilde{X}^t := \mathbf{1}(\tau \geq t) \cdot X^t$ and $\tilde{B}^t := (1 - \mathbf{1}(\tau \geq t)) \cdot Z^t = \Pr(\tilde{X}^t = 0 \mid \mathcal{F}_{t-1})$ to obtain the result in the stopped case.

In this specific proof we use bold for vectors and capital letters for random variables, respectively. Moreover, for a vector $\mathbf{v} = (v^1, \ldots, v^n)$ and $t \leq n$, we use $\mathbf{v}^{\leq t}$ (resp. $\mathbf{v}^{<t}$) to denote the vector $(v^1, \ldots, v^t)$ (resp. $(v^1, \ldots, v^{t-1})$). The same notation is employed to restrict a sequence of random variables to its t first elements.

Let $X = (X^1, X^2, \ldots, X^n)$ and let $Z = (Z^1, Z^2, \ldots, Z^n)$. Peeling off the variables in order $Z^1, X^1, Z^2, X^2, \ldots$ we have that for any fixing $\mathbf{z} = (z^0, z^1, \ldots, z^n)$ (without any independence assumption)

$$\Pr\left(X \quad = \quad 0 \text{ and } Z \quad = \quad \mathbf{z}\right) \quad = \quad \prod_{t=1}^{n} f(t, \mathbf{z}) \quad \cdot \quad \prod_{t=1}^{n} g(t, \mathbf{z})$$

where

$$\begin{aligned} f(t, \mathbf{z}) &= \Pr(X^t = 0 \mid Z^{\leq t} = \mathbf{z}^{\leq t}, X^{<t} = 0), \\ g(t, \mathbf{z}) &= \Pr(Z^t = z^t \mid Z^{<t} = \mathbf{z}^{<t}, X^{<t} = 0). \end{aligned}$$

For any history $\sigma \in \mathcal{F}_{t-1}$ up to time $t-1$ where $Z^t = z^t$ we have $\Pr(X^t = 0 \mid \sigma) = z^t$, which implies $\Pr(X^t = 0 \mid Z^{\leq t} = \mathbf{z}^{\leq t}, X^{<t} = 0) = z^t$. So we obtain $\Pr(X = 0 \text{ and } Z = \mathbf{z}) = \mathit{prod}(\mathbf{z}) \cdot \prod_{t=1}^{n} z^t$, where

$$\mathit{prod}(\mathbf{z}) = \prod_{t=1}^{n} g(t, \mathbf{z}).$$

Letting Ω be the set of all $\mathbf{z}$'s such that $\prod_{t=1}^{n} z^t \leq \alpha$ we have

$$\begin{aligned} \Pr\left(X = 0 \text{ and } \prod_{t=1}^{n} Z^t \leq \alpha\right) &\leq \alpha \sum_{\mathbf{z} \in \Omega} \mathit{prod}(\mathbf{z}) \\ &\leq \alpha \sum_{z^1} \cdots \sum_{z^n} \mathit{prod}(\mathbf{z}), \end{aligned}$$

where the sum $\sum_{z^t}$ ranges over all possible values of Z^t (recall that we assumed the probability space to be finite). Finally, we claim that the sum in the RHS equals 1: by using the definition of $\mathit{prod}(\mathbf{z})$ we get that

$$\begin{aligned} &\sum_{z^1} \cdots \sum_{z^n} \mathit{prod}(\mathbf{z}) \\ &= \sum_{z^1} \cdots \sum_{z^n} \mathit{prod}(\mathbf{z}^{<n}) \cdot g(n, \mathbf{z}) \\ &= \sum_{z^1} \cdots \sum_{z^{n-1}} \mathit{prod}(\mathbf{z}^{<n}) \left(\sum_{z^n} g(n, \mathbf{z})\right) \\ &= \sum_{z^1} \cdots \sum_{z^{n-1}} \mathit{prod}(\mathbf{z}^{<n}), \end{aligned}$$

Iterating this argument $n-1$ times gives that

$$\sum_{z^1} \cdots \sum_{z^n} \mathit{prod}(\mathbf{z}) = \sum_{z^1} \Pr(Z^1 = z^1) = 1,$$

which concludes the proof. □

With this we can bound the failure probability of the algorithm, concluding its analysis.

Lemma 3. *$\mathcal{A}_{base}$ with $\omega \geq m$ and $N \geq n$ returns* Fail *with probability at most $\frac{1}{N}$.*

Proof. Fix a hypothesis $h \in \mathcal{H}$. By taking a union bound over all hypotheses, it suffices to show that the probability that $\mathcal{A}_{\text{base}}$ fails upon receiving hypothesis h is at most $\frac{1}{N^2}$.

Notice that h is received at most once by $\mathcal{A}_{\text{base}}$: after receiving h for the first time, the algorithm either sends an example that covers h (so the Learner never resends h) or returns Fail. So let τ be the time when h is received, i.e., $H_\tau = h$ (let τ be the last round if h is never received), and let X^t be the indicator that at time t an example covering h was sent to Learner by the algorithm. As mentioned before, by the weight update of the algorithm, at round τ we have weight $W^\tau(h) = W^\tau(H_\tau) \geq 1$. Since $\omega \geq m$, the initial weights satisfies $W^0(h) \leq \frac{1}{2}$ and hence the weight increments up to round τ satisfy $\sum_{t \leq \tau} D^t(h) \geq \frac{1}{2}$. Moreover, if the algorithm fails on h we have $X^1 = \ldots = X^\tau = 0$, thus

$$\Pr(\text{fails on } h) \leq \Pr\left(X^t = 0, \forall t \leq \tau, \text{ and } \sum_{t \leq \tau} D^t(h) \geq \frac{1}{2}\right). \tag{1}$$

Let $\mathcal{F}_{t-1}$ be the σ-algebra generated by the history up to round $t-1$ plus the hypothesis at round t. By the sampling procedure, the conditional probability $Z^t := \Pr(X^t = 0 \mid \mathcal{F}_{t-1})$ that no example covering h was added in round t is

$$Z^t = (1 - D^t(h))^{4 \log N} \leq e^{-4(\log N) D^t(h)}, \tag{2}$$

recalling that $(1-x) \leq e^{-x}$ for all x. Then $\sum_{t \leq \tau} D^t(h) \geq \frac{1}{2}$ implies $\prod_{t \leq \tau} Z^t \leq e^{-2 \log N} = \frac{1}{N^2}$, and the RHS of (1) can be upper bounded

$$\Pr(\text{fails on } h) \leq \Pr\left(X^t = 0, \forall t \leq \tau \text{ and } \prod_{t \leq \tau} Z^t \leq \frac{1}{N^2}\right).$$

From Lemma 2 this can be further upper bounded by just $\frac{1}{N^2}$, and hence $\Pr(\text{fails on } h) \leq \frac{1}{N^2}$. This concludes the proof. □

Making the algorithm agnostic to the size of $\mathcal{H}$. In Theorem 1 we assume that the Teacher knows the number of hypotheses $n = |\mathcal{H}|$ of Learner. However, a guess and double strategy can be used to overcome this limitation. More concretely, let $\mathcal{A}_{\text{agno}}$ be a teaching algorithm that implements a sequence of calls to $\mathcal{A}_{\text{base}}$, where in the i-th call the parameter N is set to 2^{2^i} while ω is set to m. Moreover, for $i > 1$, the initial hypothesis for the i-th call is the one that failed in the call $i-1$. The procedure ends as soon as some call to $\mathcal{A}_{\text{base}}$ accepts h^*.

Let $t = \lceil \log \log n \rceil$. At the t-th call of $\mathcal{A}_{\text{base}}$ the parameter N is not smaller than n. Thus, it follows from Theorem 1 that this call sends $O(\mathcal{TS} \cdot \log m \cdot 2^t)$ examples and returns h^* with probability at least $1 - 1/n$. Moreover, by Lemma 1 the previous calls to $\mathcal{A}_{\text{base}}$ send $\sum_{i=1}^{t-1} O(\mathcal{TS} \log m \cdot 2^i) = O(\mathcal{TS} \log m \log n)$ examples. Therefore, we have the following result.

Theorem 2. *Consider teaching a worst-case learner in the realizable case $h^* \in \mathcal{H}$. The algorithm $\mathcal{A}_{agno}$, with probability at least $1 - \frac{1}{n}$, returns the correct hypothesis h^* and sends at most $O(\mathcal{TS} \log m \log n)$ examples.*

2.2 Improved Guarantee Based on the Quality of the Hypotheses

The next theorem shows that it is possible to obtain an improved bound when the distribution of the number of errors of the hypotheses in $\mathcal{H}$ is taken into account. For instance if only $O(1)$ hypotheses make a non-constant number of errors, then the bound on the number of examples sent is improved from $O(\mathcal{TS} \log m \log n)$ to $O(\mathcal{TS}(\log n + \log m))$.

Theorem 3. *Consider teaching a worst-case learner in the realizable case $h^* \in \mathcal{H}$. Let n_i be the number of hypotheses in $\mathcal{H}$ whose number of errors is between $[2^{2^i}, 2^{2^{i+1}})$ for $i \geq 1$, and let n_0 be the number of hypotheses with error in $[1, 4)$. Then there is an algorithm for the Teacher that with probability at least $\frac{4}{5}$ returns a correct hypothesis h^* and the number of examples sent is*

$$O(\mathcal{TS}(\mathcal{H})) \cdot \left(\log m + \sum_{i=0}^{\log \log m} 2^i \log(n_i + 1) \right)$$

In particular, this is $O(\mathcal{TS}(\mathcal{H}) \log m \log(\max_i n_i + 1))$.

The starting point is to notice that if we run $\mathcal{A}_{\text{base}}$ initialized with ω being the maximum number of errors of a hypothesis in the class, say err, then it sends at most $O(\mathcal{TS} \ \log err \cdot \log n)$ examples. Then the idea of the algorithm of Theorem 3 is the following: Instantiate copies $\mathcal{A}_1, \ldots, \mathcal{A}_{\log \log m}$ of the algorithm $\mathcal{A}_{\text{base}}$, where $\mathcal{A}_i$ is initialized with $\omega = 2^{2^i}$. Then when a hypothesis h comes from Learner, we see in which bucket $[2^{2^i}, 2^{2^{i+1}})$ its number of errors falls into, and send the hypothesis to algorithm $\mathcal{A}_i$.

Since the size of smallest teaching set for the hypotheses that fall in the same bucket is no larger than $\mathcal{TS}(\mathcal{H}, h^*)$, we can compose the guarantees from the algorithms $\mathcal{A}_i$'s to get the above guarantee (Details in Appendix B).

2.3 Non-realizable Case

We now consider the non-realizable case where the correct hypothesis h^* may not be in the Learner's class $\mathcal{H}$. Recall that in this case in each round the Learner sends a hypothesis in $\mathcal{H}$ with the smallest number of errors in the examples received so far, and the Teacher's goal is to make the Learner return a hypothesis in $\mathcal{H}$ with the smallest number of total errors over the whole set of examples $\mathcal{X}$.

We first need a generalization of the notion of teaching set. Informally, if the best hypothesis in $\mathcal{H}$ has k errors in $\mathcal{X}$, to isolate it the Teacher should send examples that certify that the other hypotheses have at least $k+1$ errors. We say that a set of examples $\mathcal{X}' \subseteq \mathcal{X}$ is a *k-extended teaching set* with respect to h^* if for each hypothesis $h \in \mathcal{H}$ with more than k errors there are at least $k+1$ examples in $\mathcal{X}'$ where h is wrong (differs from h^*). We let $\mathcal{TS}_k = \mathcal{TS}_k(\mathcal{H}, h^*)$ denote the size of the smallest k-extended teaching set w.r.t. h^*.

Notice that after the Learner receives a set of labeled examples that contain a k-extended teaching set, it returns a hypothesis with at most k total errors since such hypotheses have at most k errors in the examples received, while all other hypotheses have at least $k+1$ errors in them. If k is set to be the number of errors of the best hypothesis in $\mathcal{H}$, the Learner then returns an optimal hypothesis.

Theorem 4. *Consider teaching a worst-case learner where h^* may not belong to $\mathcal{H}$. Let k be the smallest number of errors of a hypothesis in $\mathcal{H}$. Then there is a Teacher's algorithm that with probability at least $1 - \frac{1}{m}$ returns a hypothesis that makes k errors and sends at most $O(\mathcal{TS}_k \ \log m \log(m+n))$ examples.*

The high-level idea of the algorithm is the same as in the realizable case: it tries to compute in an online fashion a k-extended teaching set of small size, but now based on an *online generalized set cover* algorithm [5], where elements may need to be covered multiple times. However, since the minimum number of errors k is unknown, the algorithm also needs to keep a lower bound on k that is given by the number of errors of the last received hypothesis over the examples already sent. The algorithm stops when it receives from the Learner a hypothesis whose total number of errors matches this lower bound. Details are provided in Appendix C.

3 Other Learner Models

In this section we show that better bounds are possible under reasonable assumptions on the way the Learner can choose the consistent hypotheses to return. We address the realizable case where $h^* \in \mathcal{H}$ and, here, we use $\mathcal{H}_t \subseteq \mathcal{H}$ to denote the set of hypotheses consistent with all the examples sent by the Teacher in the rounds $1, \ldots, t-1$ (so $\mathcal{H}_t$ is the set of possible hypotheses that Learner can send in this round t).

3.1 Smooth Transition Learners

We first consider the *smooth transition model* where we further assume that the Learner sends a hypothesis "close" to the one sent in the previous round. Concretely, we use the number of disagreements between hypotheses $d(h, h') = |\{x \in \mathcal{X} \mid h(x) \neq h'(x)\}|$ as measure of closeness, and assume the hypothesis h_t that Learner sends at round t is one in $\mathcal{H}_t$ with $(1+\alpha)$-approximate minimum distance to the hypothesis h_{t-1} sent in the previous round, namely

$$d(h_t, h_{t-1}) \leq (1+\alpha) \min_{h \in \mathcal{H}_t} d(h, h_{t-1}).$$

We provide an algorithm $\mathcal{A}^{\alpha}_{\text{close}}$ for this model whose guarantee depends on the number of errors err_1 of the first hypothesis sent by the Learner. That means that if the Learner has a good guess for the right hypothesis, fewer examples are needed to complete the teaching. Algorithm $\mathcal{A}^{\alpha}_{\text{close}}$ is obtained from $\mathcal{A}_{\text{base}}$ via two simple modifications: The starting weights of the examples W^0_e are set to $1/2err_1$ instead of $1/2m$, and the number of examples sampled per round is $\frac{8}{1-2\alpha} \log N$ instead of $4 \log N$. This algorithm has the following guarantee.

Theorem 5. *Consider the smooth transition model with $\alpha \in [0, \frac{1}{2})$ in the realizable case $h^* \in \mathcal{H}$. Let err_1 be the number of errors over $\mathcal{X}$ of the initial hypothesis sent by Learner. Then algorithm $\mathcal{A}^{\alpha}_{close}$ set with $N \geq n$ sends $O(\mathcal{TS}\frac{1}{1-2\alpha} \log err_1 \log N)$ examples and with probability at least $1 - \frac{1}{N}$ returns the correct hypothesis h^*.*

The main observation for obtaining a guarantee that depends on err_1 is that in the smooth transition model the number of errors of the hypotheses sent by the Learner cannot increase rapidly. We have the following.

Lemma 4. *Let h, h' be the hypotheses returned by Learner at rounds $t-1$ and t respectively. Then:*

a) $|wrong(h')| \leq 2\,|wrong(h') \cap wrong(h)| + \alpha\,|wrong(h)$

b) $|wrong(h')| \leq (4 + 2\alpha)\, err_1$.

Proof. We first prove item (a). Let wrong$(h \setminus h')$ (resp. wrong$(h' \setminus h)$) be the set of examples where only h (resp. h') is wrong. In addition, let $DIFF$ (resp. EQ) be the number of examples that both h and h' are wrong but give give different (resp. equal) classification. In formulae,

$$\begin{aligned} DIFF &= \{e \in \mathrm{wrong}(h) \cap \mathrm{wrong}(h') \mid h(e) \neq h'(e)\} \\ EQ &= \{e \in \mathrm{wrong}(h) \cap \mathrm{wrong}(h') \mid h(e) = h'(e)\}. \end{aligned}$$

The number of disagreements between these hypotheses is

$$d(h, h') = |\mathrm{wrong}(h \setminus h')| + |\mathrm{wrong}(h' \setminus h)| + DIFF$$

$$d(h, h^*) = |\mathrm{wrong}(h \setminus h')| + DIFF + EQ$$

The smooth transition model guarantees that $d(h, h') \leq (1+\alpha)\, d(h, h^*)$ so that $|\mathrm{wrong}(h' \setminus h)| \leq \alpha\, |\mathrm{wrong}(h' \setminus h)| + (1 + \alpha)\, |\mathrm{wrong}(h) \cap \mathrm{wrong}(h')|$, and hence

$$\begin{aligned} |\mathrm{wrong}(h')| = |\mathrm{wrong}(h' \setminus h)| + |\mathrm{wrong}(h) \cap \mathrm{wrong}(h')| \\ \leq \alpha\, |\mathrm{wrong}(h \setminus h')| \\ + (2 + \alpha)\, |\mathrm{wrong}(h) \cap \mathrm{wrong}(h')| \\ = \alpha\, |\mathrm{wrong}(h)| + 2|\mathrm{wrong}(h) \cap \mathrm{wrong}(h')|, \end{aligned}$$

which establishes item (a).

Proof of item [b]. If h' is the first hypothesis, the result clearly holds because the first hypothesis makes err_1 mistakes.

Thus, let $t > 1$ be the round in which h' is received and let h be hypothesis received at the round $t - 1$. We have that $|\mathrm{wrong}(h)| < 2err_1$, for otherwise h would have weight at least 1 by the beginning of round $t - 1$ and, hence, $t - 1$ would be the last round of the interaction. Thus, it follows from item (a) and from $|\mathrm{wrong}(h) \cap \mathrm{wrong}(h') \leq |\mathrm{wrong}(h)|$ that $|\mathrm{wrong}(h')| \leq (4 + 2\alpha)err_1$. □

Proof of Theorem 5. The bound on the number of examples follows directly from Lemma 1, since $\mathcal{A}^{\alpha}_{\mathrm{close}}$ behaves as $\mathcal{A}_{\mathrm{base}}$ initialized with $\omega = err_1$, but sending $\frac{2}{1-\alpha}$ times as many examples per round.

The proof that the probability of returning the correct hypothesis is at least $1 - \frac{1}{N}$ is similar to that of Lemma 3: we need to show that if the algorithm receives hypothesis h' on round τ then $\sum_{t \leq \tau} D^t(h')$, the total increase of the weights of the wrong examples of a hypothesis h' is "large". That is enough since the concentration arguments following inequality (1), then, guarantee that the probability of failing at this point (i.e., no examples covering h' were sent) is small. More precisely, it suffices to show

$$\sum_{t \leq \tau} D^t(h') \geq \frac{1 - 2\alpha}{4}. \tag{3}$$

We note that in Lemma 3 we have the stronger lower bound with RHS $\frac{1}{2}$; the difference is compensated by the extra number of examples sent in each round by $\mathcal{A}^{\alpha}_{\mathrm{close}}$. We prove inequality (3) by considering two cases:

Case 1. $|\mathrm{wrong}(h')| \leq err_1$. In this case $\sum_{t \leq \tau} D^t(h')$ is at least $\frac{1}{2} > \frac{1-2\alpha}{4}$ since the initial weight of h' is at most $\frac{1}{2}$ and its final weight is at least 1 by the weight update step of the algorithm.

Case 2. $|\mathrm{wrong}(h')| > err_1$. It follows from item (a) of Lemma 4 that the hypothesis received at round $\tau - 1$, say h, shares at least $(|\mathrm{wrong}(h')| - \alpha\, |\mathrm{wrong}(h)|)/2$ wrong examples with h'.

Moreover, we have $|\text{wrong}(h)| \leq 2err_1$: otherwise the starting weight of this hypothesis $W^0(h)$ is already at least 1, so the algorithm fails on round $\tau - 1$, contradicting that it fails on round τ. Together with the assumption from being in Case 2, this gives $|\text{wrong}(h)| \leq 2|\text{wrong}(h')|$ and hence the number of common wrong examples between h and h' is at least $\frac{1-2\alpha}{2}\,|\text{wrong}(h')|$. Since the weight of each of these common examples was increased by at least $1/2err_1$ in round $\tau - 1$ (weights are doubled and start at $1/2err_1$), we have that

$$\sum_{t \leq \tau} D^t(h') \geq D^{\tau-1}(h') \geq \frac{1-2\alpha}{2} \cdot err_1 \cdot \frac{1}{2err_1} = \frac{1-2\alpha}{4}.$$

This proves (3) and concludes the proof of Theorem 5.

Making the algorithm agnostic to the size of $\mathcal{H}$. To obtain an algorithm agnostic to the size of $\mathcal{H}$ we proceed as in Section 2.1 but performing a sequence of calls to $\mathcal{A}^{\alpha}_{\text{close}}$, rather than to $\mathcal{A}_{\text{base}}$.

The only different issue that arises in the analysis of this algorithm is how to bound the number of errors $err_{1,i}$ made by the first hypothesis of ith call of $\mathcal{A}^{\alpha}_{\text{close}}$. By using the fact that this hypothesis is exactly the last one returned by the previous call together with item (b) of Lemma 4, we get that $err_{1,i} \leq (4+2\alpha)\, err_{1,i-1}$ and, hence, $err_{1,i} \leq (4+2\alpha)^{i-1}\, err_{1,1}$. This observation together with the same arguments employed in the analysis of $\mathcal{A}_{\text{agno}}$ allows us to establish the following theorem:

Theorem 6. *Under the same assumptions as in Theorem 5, there is a teacher's algorithm agnostic to the number of hypothesis n that sends $O(\mathcal{TS}\, \log n\, (\log err_1 + \log\log n))$ examples and with probability at least $1 - \frac{1}{n}$ returns the correct hypothesis h^*.*

Note that in the worst-case learner model this bound is not achievable by poly-time algorithms unless $\mathcal{NP} \subset \mathcal{BPP}$. This is shown through a simple modification of a lower bound from [14] [See Appendix D]

3.2 The Random Learner Model

We assume that in each round the Learner sends a batch of *random* i.i.d. hypotheses from the ones that are consistent with the examples received thus far.

We show that in this situation the Teacher can exploit the randomness of the Learner's choice in order to estimate the example that covers (i.e., falsifies) the highest number of hypotheses (which are consistent with the examples seen so far). With this knowledge, the Teacher can resort to algorithms for the *offline* set cover problem and significantly improve the number of examples used: we show that Teacher sends with high probability $O(\mathcal{TS}\, \log(n+m))$ examples, which is the best bound achievable in polytime, under the assumption that $\mathcal{P} \neq \mathcal{NP}$, for the relevant case where the number of hypotheses n is larger than the number of examples m [21].

Algorithm $\mathcal{A}_{\text{rand}}$ (Figure 2) runs the greedy approximation algorithm for offline set cover over the empirical process: At each round t the Teacher requests a batch $\widetilde{\mathcal{H}}_t$ of T random hypotheses from the set $\mathcal{H}_t$ of hypotheses consistent with the examples sent thus far, and sends to Learner an example $\tilde{e}$ that covers the largest number of hypotheses from $\widetilde{\mathcal{H}}_t$. The size T of the requested batch ideally depends on the size of the smallest teaching set $\mathcal{TS}(\mathcal{H}, h^*)$, but since this quantity is unknown the algorithm also employs a guess-and-double approach: In phase i it uses $T = 2^i$ as a guess for $\mathcal{TS}$ and runs the greedy procedure for $2T$ rounds. If within these rounds the algorithm does not terminate, the phase is concluded and phase $i+1$ is started.

The following theorem is the main result of this section.

```
Algorithm $\mathcal{A}_{\text{rand}}$
Input: Examples $\mathcal{X}$
  1. Initialize round counter $t = 0$
  2. For each phase $i = 1, 2, \ldots$:
      • Update the teaching set size guess $T = 2^i$
      • For $2T$ rounds
          – Update round counter $t = t + 1$
          – Receive batch $\widetilde{\mathcal{H}}_t$ of $T$ random hypotheses from $\mathcal{H}_t$
          – If all hypotheses in $\widetilde{\mathcal{H}}_t$ equal $h^*$, then Return    (*)
          – Send $\tilde{e} = \text{argmax}_e \, |\{h \in \widetilde{\mathcal{H}}_t \mid e \in \text{wrong}(h)\}|$
```

Figure 2: Teacher $\mathcal{A}_{\text{rand}}$

Theorem 7. *Consider the random learner model in the realizable case $h^* \in \mathcal{H}$. With probability at least $1 - O(1/m)$, $\mathcal{A}_{rand}$ satisfies the following: (i) it accepts the target hypothesis h^* (line* **(*)** *of $\mathcal{A}_{rand}$); (ii) it sends $O(\mathcal{TS} \cdot \log(n+m))$ examples and (iii) it receives $O(\mathcal{TS} \cdot \log(n+m))$ hypotheses per round.*

We note that without the bound on the number of hypotheses received per round the result would be straightforward since the Teacher could request infinitely many hypotheses to acquire a very accurate knowledge of the class $\mathcal{H}$ and then resort to the off-line greedy set cover algorithm.

To prove Theorem 7, given a set of hypotheses $\mathcal{H}' \subseteq \mathcal{H}$ let $c^*(\mathcal{H}')$ be the maximum number of hypotheses from $\mathcal{H}'$ that can be covered by a single example in $\mathcal{X}$. We rely on two lemmas whose proofs can be found in Appendix E. The first shows that with high probability the example that covers the largest number of hypotheses in the empirical set $\widetilde{\mathcal{H}}_t$ also covers a large number of hypotheses from $\mathcal{H}_t$.

Lemma 5. *Let $\tilde{e}$ be an example that covers the largest number of hypotheses in $\tilde{\mathcal{H}}_t$ and let $c_{\tilde{e}}$ be the number of hypotheses* covered by *$\tilde{e}$ in $\mathcal{H}_t$. If $|\widetilde{\mathcal{H}}_t| \geq 40\mathcal{TS} \ln m$ then,*

$$\Pr\left(c_{\tilde{e}} \leq \frac{1}{12} c^*(\mathcal{H}_t)\right) \leq \frac{2}{m^4}.$$

The second lemma gives an upper bound on the number of rounds executed by an approximate version of the greedy offline set cover algorithm.

Lemma 6. *Consider $0 < \alpha < 1$ and $\mathcal{A}_\alpha$ be any teacher's algorithm that at each round t sends to Learner an example that covers at least $\alpha \cdot c^*(\mathcal{H}_t)$ hypotheses from $\mathcal{H}_t$. Then $\mathcal{A}_\alpha$ executes $O(\frac{1}{\alpha}\mathcal{TS} \ln n) = O(\mathcal{TS} \ln n)$ rounds before the only consistent hypothesis is h^*, i.e., $\mathcal{H}_t = \{h^*\}$.*

Proof of Theorem 7. Let $\alpha = 1/12$ and let $\hat{i}$ be the first phase where $T = 2^{\hat{i}}$ is at least $\mathcal{TS} \cdot \max\{\frac{1}{\alpha} \ln n, 40 \ln m\}$. Since the previous $\hat{i} - 1$ phases send $\sum_{j=1}^{\hat{i}-1} 2^j = O(\mathcal{TS} \ln(m+n))$ examples and each of them requests $O(\mathcal{TS} \ln(n+m))$ hypotheses per round, it suffices to analyse phase $\hat{i}$ onwards.

We say that an example is *bad* for round t if it does not cover $\alpha c^*(\mathcal{H}_t)$ hypothesis of $\mathcal{H}_t$. Due to Lemma 5 every round t that occurs after the beginning of phase $\hat{i}$ sends a bad example with probability at most $2/m^4$. Thus, it follows from the union bound that a bad example is sent during the $\frac{1}{\alpha}\mathcal{TS} \log n$ first rounds of phase $\hat{i}$ with probability at most $(2\frac{1}{\alpha}\mathcal{TS} \log n)/m^4 \leq 24/m$, where the inequality holds because $\mathcal{TS} \leq m$ and $n \leq |\mathcal{Y}|^m \leq m^m$. Hence, it follows by Lemma 6 that, with probability at least $1 - O(1/m)$, the only consistent hypothesis that remains after these round is h^*. Since each of these rounds requests $O(\mathcal{TS}\log(n+m))$ hypotheses, the theorem is proved. □

4 Non-redundant Teaching Sets

We say that a teaching set X is *redundant* if it contains a redundant example, that is, an example e such that $X \setminus \{e\}$ is still a teaching set. The algorithms discussed so far may construct teaching sets that are redundant. We show that $|X| \cdot (|\mathcal{Y}| - 1)$ additional rounds suffice to obtain a non-redundant teaching set from a teaching set X w.r.t. $(h^*, \mathcal{H})$, where $\mathcal{Y}$ is the set of possible labels for the examples.

For that we consider a more general interaction model where at each round Teacher sends a set of labelled examples to Learner and the latter returns a hypothesis that makes the smallest number of errors in this set (ignoring the examples received at previous rounds). Differently from the previous sections, here the Teacher may send an example e with label different from $h^*(e)$.

The following proposition gives a simple condition for deciding whether e is redundant for teaching set X.

Proposition 1. *Let X be a teaching set for $(\mathcal{H}, h^*)$. If there exists a hypothesis $h \in \mathcal{H}$ with $h(e') = h^*(e')$ for every $e' \in X \setminus \{e\}$ and $h(e) \neq h^*(e)$, then the example e is non-redundant for the set X, and also for any teaching set contained in X. Otherwise, e is redundant for X.*

Given this observation, the algorithm for obtaining a non-redundant teaching set from X is straightforward: It scans the examples in X and for each $e \in X$ verifies whether e is redundant (w.r.t. the set of examples that have not been removed from X) or not; if it is, the example is removed from X and the scan continues over the examples that have not been tested yet. To verify whether e is redundant the Teacher interacts with the Learner in $|\mathcal{Y}| - 1$ rounds testing the existence of a label $y \neq h^*(e)$ for which the Learner returns a hypothesis consistent with the labelled set of examples $\mathcal{D}_y = \{(e', h^*(e')) \mid e' \in X \setminus e\} \cup \{(e, y)\}$. If such label does not exist then the algorithm concludes that e is redundant. It is clear that the overall number of rounds is at most $|X| \cdot (|\mathcal{Y}| - 1)$.

In simulations on synthetic data (Appendix F) this method significantly reduced the teaching sets found by $\mathcal{A}_{\text{agno}}$.

5 Computational Experiments

Although our work is mainly theoretical we performed experiments to understand how our Teacher $\mathcal{A}_{\text{agno}}$ compares with a non interactive one over real datasets.

The non interactive teacher, denoted by `NIT`, receives an integer ℓ and then sends to the Learner ℓ randomly selected examples. We compared the number of examples that both $\mathcal{A}_{\text{agno}}$ and `NIT` need to send in order to attain a certain level of accuracy. For our evaluation we used Random Forest and Light Gradient Boosting Machine (LGBM) as learners, and conducted experiments on 12 datasets: `mnist` and 11 others from the UCI repository (`mushroom`, `avila`, `bank_marketing`, `car`, `Credit Card`, `Firm_Teacher`, `crowdsourced`, `Electrical_grid`, `HTRU`, `nursery` and `Sensorless_drive`).

Table 1 shows the number of examples (relative to the size of the full dataset) required by each pair Teacher-Learner to attain an accuracy over the full dataset larger than $z\%$ of that obtained when the Learner is trained/tested over the full dataset. Each numeric entry of this table is an average of 12 values, where each of them corresponds to a distinct dataset. More details are presented in Appendix G.

Table 1: Percentage of the size of the full dataset required by each Teacher-Learner to achieve an accuracy larger than $z\%$ (with $z \in \{90, 95, 99\}$) of that achieved by the Learner when it is trained and tested in the full dataset.

Teacher-Learner	90%	95%	99%
$\mathcal{A}_{\text{agno}}$-Random Forest	10.1%	14.7%	20.6 %
NIT-Random Forest	14.7%	30.4%	59.9%
$\mathcal{A}_{\text{agno}}$-LGBM	2.8 %	5.7 %	8.9 %
NIT-LGBM	3.4 %	7.8 %	28.0 %

The results provide evidence that $\mathcal{A}_{\text{agno}}$ requires significantly fewer examples than NIT (e.g. a factor of 3 for the target 99%). Furthermore, an interesting observation is that the advantage of $\mathcal{A}_{\text{agno}}$ increases as the level of accuracy requested gets higher. A reasonable explanation is that when little is known (low accuracy in our setting), most of the examples are useful (hence, random sampling also works well); however, when a certain level of knowledge has already been reached, more specific examples, as those provided by $\mathcal{A}_{\text{agno}}$, are needed to further increase it.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments including also some pointers to very recent related literature. We would also like to thank Daniel dos Santos Marques for the fruitful discussions about applications of machine teaching and to Sanjoy Dasgupta for clarifications about the paper "Teaching a black-box learner".

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

The work of the third author is partially supported by CNPq under grant 307572/2017- 0 and by FAPERJ, grant Cientista do Nosso Estado E- 26/202.823/2018.

The work of the of the fourth author is supported by CNPq grants Universal #431480/2016-8 and Bolsa de Produtividade 35 em Pesquisa #4310516/2017-0, FAPERJ grant Jovem Cientista do Nosso Estado.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor, *The online set cover problem*, SIAM J. Comput, 39 (2009), pp. 361–370.

[2] D. Angluin and T. Dohrn, *The power of random counterexamples*, Theor. Comput. Sci., 808 (2020), pp. 2–13.

[3] Anonymous and S. Dasgupta. Private Communication, Jan.

[4] F. J. Balbach and T. Zeugmann, *Teaching randomized learners with feedback*, Inf. Comput, 209 (2011), pp. 296–319.

[5] N. Buchbinder and J. S. Naor, *Online primal-dual algorithms for covering and packing*, Math. Oper. Res., 34 (2009), pp. 270–286.

[6] Y. Chen, A. Singla, O. Mac Aodha, P. Perona, and Y. Yue, *Understanding the role of adaptivity in machine teaching: The case of version space learners*, in Advances in Neural

Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., 2018, pp. 1476–1486.

[7] S. Dasgupta, D. Hsu, S. Poulis, and X. Zhu, *Teaching a black-box learner*, in Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, K. Chaudhuri and R. Salakhutdinov, eds., vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 1547–1555.

[8] V. H. de la Peña and E. Giné, *Decoupling: From Dependence to Independence*, Probability and Its Applications, Springer New York, 1999.

[9] D. A. Freedman, *On tail probabilities for martingales*, Annals of Probability, 3 (1975), pp. 100–118.

[10] Z. Gao, C. Ries, H. U. Simon, and S. Zilles, *Preference-based teaching*, J. Mach. Learn. Res, 18 (2017), pp. 31:1–31:32.

[11] S. A. Goldman and M. J. Kearns, *On the complexity of teaching*, J. Comput. Syst. Sci, 50 (1995), pp. 20–31.

[12] E. Johns, O. M. Aodha, and G. J. Brostow, *Becoming the expert - interactive multi-class machine teaching*, in CVPR, IEEE Computer Society, 2015, pp. 2616–2624.

[13] D. Kirkpatrick, H. U. Simon, and S. Zilles, *Optimal collusion-free teaching*, in Proceedings of the 30th International Conference on Algorithmic Learning Theory, A. Garivier and S. Kale, eds., vol. 98 of Proceedings of Machine Learning Research, Chicago, Illinois, 22–24 Mar 2019, PMLR, pp. 506–528.

[14] S. Korman, *On the Use of Randomization in the Online Set Cover Problem*, PhD thesis, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Israel, 2004.

[15] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song, *Iterative machine teaching*, in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, D. Precup and Y. W. Teh, eds., vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 2149–2158.

[16] W. Liu, B. Dai, X. Li, Z. Liu, J. M. Rehg, and L. Song, *Towards black-box iterative machine teaching*, in ICML, J. G. Dy and A. K. 0001, eds., vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 3147–3155.

[17] F. Mansouri, Y. Chen, A. Vartanian, J. Zhu, and A. Singla, *Preference-based batch and sequential teaching: Towards a unified view of models*, in Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 9195–9205.

[18] S. Mei and X. Zhu, *Using machine teaching to identify optimal training-set attacks on machine learners*, in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI?15, AAAI Press, 2015, p. 2871?2877.

[19] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.

[20] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto, *Faster teaching via pomdp planning*, Cognitive Science, 40 (2016), pp. 1290–1332.

[21] Raz and Safra, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, in STOC: ACM Symposium on Theory of Computing (STOC), 1997.

[22] A. Shinohara, *Teachability in computational learning*, New Generation Comput, 8 (1991), pp. 337–347.

[23] A. Singla, I. Bogunovic, G. Bartók, A. Karbasi, and A. Krause, *Near-optimally teaching the crowd to classify*, in ICML, vol. 32 of JMLR Workshop and Conference Proceedings, JMLR.org, 2014, pp. 154–162.

[24] Y. Zhou, A. R. Nelakurthi, and J. He, *Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners*, in KDD, Y. Guo and F. Farooq, eds., ACM, 2018, pp. 2817–2826.

[25] X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty, *An overview of machine teaching*, CoRR, abs/1801.05927 (2018).

[26] S. Zilles, S. Lange, R. Holte, and M. Zinkevich, *Models of cooperative teaching and learning*, J. Mach. Learn. Res, 12 (2011), pp. 349–384.

Appendix

A On the Proof of Lemma 5 from [7]

The teaching algorithm of [7] is similar to algorithm $\mathcal{A}_{\text{base}}$ presented in Section 2, the only difference being in the sampling step: the former sends example x to the learner when its weight crosses a random threshold T_x that is sampled at the beginning of the algorithm from an exponential distribution with rate $\lambda = \ln(N/\delta)$, where N is an upper bound on $|\mathcal{H}|$ and the parameter δ is the failure probability of the algorithm.

The Lemma 5 of [7] states that the failure probability of their algorithm is at most δ. The proof fixes a hypothesis h and then considers the first point in time where its weight is at least 1. Then, the proof claims that

$$Pr[h \text{ is not covered } \mid \text{ weight of } h \geq 1] = \prod_{x \in wrong(h)} Pr[w(x) \leq T_x] = \tag{4}$$

$$\prod_{x \in wrong(h)} exp(-\lambda w(x)) \leq exp(-\lambda) = \frac{\delta}{N}. \tag{5}$$

Finally, the proof takes the union bound over all hypotheses to obtain the bound δ.

First, it was not clear whether the weights $w(x)$ are deterministic or stochastic in the above equations. Moreover, in either case it is also not clear how the first equality of (4) could be obtained. In particular, if the weights are deterministic, which makes more sense given the following equalities, (4) seems to be ignoring the fact that the conditioning (weight of $h \geq 1$) affects the distribution probability of the thresholds and, apparently, is also assuming an independence among events that does not seem to be valid.

As kindly explained in [3], for the sake of analysis, one should take into account that the probability of not sending example x in round t is given by

$$Pr[T_x > W_x^t | T_x > W_x^{t-1}] = Pr[T_x > W_x^t - W_x^{t-1}] = Pr[T_x > D_x^t] = exp(-\lambda \cdot D_x^t).$$

Although this insight is indeed very useful to clarify how the algorithm could be analysed, it does not help to overcome the issue about the lack of independence among rounds discussed right after Lemma 1.

In this sense, by using this insight and our technical lemma about adapted Bernoulli's (Lemma 2) we explain how to derive a formal proof for Lemma 5 from [7].

A proof similar to that of Lemma 3 can be used to establish Lemma 5, with the only difference being on how the bound on $Z^t = Pr[X^t = 0|\mathcal{F}_{t-1}]$, given by inequality (2), is derived.

In what follows we prove that

$$Pr[X^t = 0|\mathcal{F}_{t-1}] \leq exp(-\lambda \cdot D^t(h)).$$

First note that

$$Z^t = Pr[X^t = 0|\mathcal{F}_{t-1}] = Pr[W_x^t < T_x \text{ for all } x \in wrong(h)|\mathcal{F}_{t-1}]$$

Recall that no example in $wrong(h)$ is sent during history $\mathcal{F}_{t-1}$. Thus, if a point $\mathbf{t} = (T_1, \ldots, T_m)$ in the thresholds' probability space leads to history $\mathcal{F}_{t-1}$ then $T_x > W_x^{t-1}$ for all $x \in wrong(h)$, where W_x^{t-1} is the weight of example x by the end of history $\mathcal{F}_{t-1}$. Moreover, assuming a deterministic learner, every point $\mathbf{t}' = (T'_1, \ldots, T'_m)$ simultaneously satisfying:

- $T'_x > W_x^{t-1}$ for all $x \in wrong(h)$

- $T'_x = T_x$ for all $x \notin wrong(h)$

leads to the same history. Therefore,

$$\begin{aligned} &Pr[W_x^t < T_x \text{ for all } x \in wrong(h) | \mathcal{F}_{t-1}] = \\ &Pr[W_x^t < T_x \text{ for all } x \in wrong(h) | W_x^{t-1} < T_x \text{ for all } x \in wrong(h)] \leq \\ &\prod_{x \in wrong(h)} exp(-\lambda \cdot (W_x^t - W_x^{t-1})) = exp(-\lambda \cdot D^t(h)) \end{aligned}$$

B Improved guarantee based on the quality of the hypotheses

Let $\mathcal{A}_{\text{base}}(N, \omega)$ denote the teacher's algorithm from Figure 1, making its inputs explicit. To obtain the guarantee of Theorem 3 we refine this base algorithm in a couple of steps.

First, Lemmas 1 and 3 actually give the following guarantee for $\mathcal{A}_{\text{base}}(N, \omega)$, even in the case when $\mathcal{H}$ does not contain a correct hypothesis, i.e., $h^* \notin \mathcal{H}$. Notice that this case the interaction between the Learner and $\mathcal{A}_{\text{base}}(N, \omega)$ stops because of one of two events:

1. $\mathcal{A}_{\text{base}}(N, \omega)$ returns FAIL
2. The Learner stops sending hypotheses, because there are no hypotheses $\mathcal{H}$ consistent with the examples provided by the Teacher

In contrast, when there is a correct hypothesis in $\mathcal{H}$ the interaction stops when $\mathcal{A}_{\text{base}}(N, \omega)$ returns either FAIL or a correct hypothesis.

Theorem 8. *Consider a hypothesis class $\mathcal{H}$ and examples $\mathcal{X}$, possibly with no hypothesis in the class that is correct on all examples. Suppose all hypotheses in $\mathcal{H}$ have at most ω errors. Then the Teacher's algorithm $\mathcal{A}_{base}(N, \omega)$ always sends at most*

$$\mathcal{TS}(\mathcal{H}) \cdot O(\log w \log N)$$

examples. Moreover, if $N \geq |\mathcal{H}|$ then with probability at least $1 - \frac{1}{N}$:

- *If there is a classifier in $\mathcal{H}$ that is correct in all examples $\mathcal{X}$, $\mathcal{A}_{base}(N, \omega)$ returns one such classifier*
- *Else, the interaction between the Learner and $\mathcal{A}_{base}(N, \omega)$ stops because the former stopped sending hypotheses*

Algorithm agnostic to the number of hypotheses. The next step in the construction of our final algorithm is to convert the algorithm $\mathcal{A}_{\text{base}}(N, \omega)$ into one that does not need an explicit estimate N of the number of hypotheses in the class $\mathcal{H}$; this is done using a simple "guess-and-double" approach. Actually it will be convenient to be able to provide a "starting estimate" N^0 to ensure that the algorithm fails with probability at most $\frac{1}{N^0}$ (instead of $\frac{1}{|\mathcal{H}|}$), regardless of the size of $\mathcal{H}$.

Theorem 9. *Consider the assumptions from Theorem 8. Consider $N^0 \geq 1$ and let $\bar{n} := \max\{|\mathcal{H}|, N^0\}$. There is an event G that holds with probability at least $1 - \frac{1}{\bar{n}}$ such that under G the Teacher's algorithm $\mathcal{A}_{agno}(N^0, w)$ sends at most*

$$\mathcal{TS}(\mathcal{H}) \cdot O(\log w \log \bar{n})$$

examples. Moreover, if there is a hypothesis in $\mathcal{H}$ that is correct in all examples $\mathcal{X}$, under G the algorithm returns one such hypothesis.

```
Algorithm $\mathcal{A}_{\text{agno}}$
Input: Examples $\mathcal{X}$, starting estimate $N^0 \geq 1$ of the number of hypothesis, initial weight $\omega \geq 0$
Let $i_0$ be the smallest integer $i$ such that $2^{2^i} \geq N^0$
For phases $i = i_0, i_0 + 1, \ldots$
    • Interact with the Learner by executing $\mathcal{A}_{\text{base}}(2^{2^i}, \omega)$ until the interaction stops
    • If $\mathcal{A}_{\text{base}}(2^{2^i}, \omega)$ returned FAIL, move to the next phase
    • If $\mathcal{A}_{\text{base}}(2^{2^i}, \omega)$ returned a correct classifier, then simply return it
    • If the interaction stopped because the Learner did not send any additional hypotheses, return
      NO CORRECT HYPOTHESIS
```

Figure 3: Algorithm $\mathcal{A}_{\text{agno}}$

Proof. Let $\bar{i}$ be the smallest integer i such that $2^{2^i} \geq \bar{n}$, and let G be the event that $\mathcal{A}_{\text{agno}}$ stops by phase $\bar{i}$. If $\mathcal{A}_{\text{agno}}$ reaches phase $\bar{i}$, it runs $\mathcal{A}_{\text{base}}(N, \omega)$ with $N \geq \bar{n} \geq |\mathcal{H}|$ and hence the guarantee from Theorem 8 holds. In this case, with probability at least $1 - \frac{1}{\bar{n}}$ the algorithm $\mathcal{A}_{\text{agno}}(N^0, w)$ stops in this phase $\bar{i}$, either by returning a correct hypothesis in $\mathcal{H}$ (if one exists), or otherwise by returning NO CORRECT HYPOTHESIS. Thus, the event G of stopping by phase $\bar{i}$ happens with probability at least $1 - \frac{1}{\bar{n}}$.

To bound the number of examples sent by $\mathcal{A}_{\text{agno}}$ under the event G we use Theorem 8 in each phase $i \leq \bar{i}$ to obtain

$$\sum_{i \leq \bar{i}} \mathcal{TS}(\mathcal{H}) \cdot O(\log w \log 2^{2^i}) = O(\mathcal{TS}(\mathcal{H}) \log w) \sum_{i \leq \bar{i}} 2^i \leq O(\mathcal{TS}(\mathcal{H}) \log w \log \bar{n}),$$

as desired.

Finally, when there is a correct hypothesis in $\mathcal{H}$ the algorithm does not return NO CORRECT HYPOTHESIS and hence under G it returns a correct hypothesis (by phase $\bar{i}$). This concludes the proof. □

Improved algorithm. We can now finally present algorithm that gives Theorem 3. We assume WLOG that $\log \log m$ is an integer, otherwise we simply pad the input with additional dummy examples. Define the intervals $B_0 = [1, 2)$ and $B_i = [2^{2^{i-1}}, 2^{2^i})$ for $i = 1, \ldots, \log \log m$.

We show that this algorithm satisfies the guarantees of Theorem 3.

Proof of Theorem 3. Let E be the event that the bound from Theorem 9 holds for all algorithms $\mathcal{A}_0, \ldots, \mathcal{A}_{\log \log m}$. By a union bound, the probability that E does not hold is at most

$$\sum_{i=0}^{\log \log m} \frac{1}{N_i^0} = \frac{1}{10} \cdot \sum_{j=1}^{\log \log m + 1} \frac{1}{j^2} \leq \frac{1}{10} \cdot \sum_{j=1}^{\infty} \frac{1}{j^2} = \frac{1}{10} \cdot \frac{\pi^2}{6} \leq \frac{1}{5}.$$

Thus, it suffices to show that whenever E holds the algorithm $\mathcal{A}_{\text{err}}$ returns a correct hypothesis and the number of examples sent is upper bounded by the desired quantity.

For that, let $\mathcal{H}_i$ be the set of hypotheses in $\mathcal{H}$ whose number of errors belongs to the interval B_i and let T_i be the set of time steps where a hypothesis was sent to algorithm $\mathcal{A}_i$. Restricting to the rounds in T_i, we have an execution of $\mathcal{A}_i$ over an adaptive sequence of hypothesis in $\mathcal{H}_i$. By definition, none of the $\mathcal{H}_i$'s contains a correct classifier. Therefore, under the event E Theorem 9 guarantees that the execution of each $\mathcal{A}_i$ stops after it sends at most

```
Algorithm $\mathcal{A}_{\text{err}}$
Input: Examples $\mathcal{X}$
1. Define the values $N_i^0 := 10(\log\log m - i + 1)^2$
2. Initialize algorithms $\mathcal{A}_0 := \mathcal{A}_{\text{agno}}(N_0^0, 2)$ and $\mathcal{A}_i := \mathcal{A}_{\text{agno}}(N_i^0, 2^{2^i})$ for $i = 1, \ldots, \log\log m$
3. For each time step:
    • Receive hypothesis $h$ from Learner. If $h$ has no errors, return it
    • Let $\bar{i}$ be such that $|\text{wrong}(h)| \in B_{\bar{i}}$
    • Send hypothesis $h$ to $\mathcal{A}_{\bar{i}}$, and receive a set of examples in $\mathcal{X}$ (If receives FAIL, return FAIL)
    • Send the set of examples to Learner
```

Figure 4: Algorithm $\mathcal{A}_{\text{err}}$

$O(\mathcal{TS}(\mathcal{H}_i) \log 2^{2^i} \log \bar{n}_i)$ examples, because the learner does not send any more hypotheses from $\mathcal{H}_i$ (where $\bar{n}_i := \max\{|\mathcal{H}_i|, N_i^0\}$). Let $I \subseteq \{0, \ldots, \log\log m\}$ be the set of i's such that $\mathcal{H}_i$ is non-empty, and notice that for indices $i \notin I$ the algorithm $\mathcal{A}_i$ is never evoked. Thus, under E, after at most

$$\sum_{i \in I} O(\mathcal{TS}(\mathcal{H}_i) \cdot 2^i \log \bar{n}_i) \tag{6}$$

examples the Learner must send a hypothesis in $\mathcal{H} \setminus \bigcap_i \mathcal{H}_i$, namely a correct hypothesis, in which case the algorithm $\mathcal{A}_{\text{err}}$ will return it.

To conclude the proof we just need to perform some algebraic manipulations to further upper bound (6). To simplify the notation let $n_i := |\mathcal{H}_i|$. Since the function $x \mapsto \log(x+1)$ is subadditive over the non-negative reals, we have that for every $a \geq 0$ and $b \geq 1$

$$\log(\max\{a, b\}) \leq \log(a + (b-1) + 1) \overset{subadd}{\leq} \log(a+1) + \log b.$$

Applying this to $\log \bar{n}_i$ and using the monotonicity of the teaching set, which gives $\mathcal{TS}(\mathcal{H}_i) \leq \mathcal{TS}(\mathcal{H})$, the number of examples under E given by (6) is at most

$$O(\mathcal{TS}(\mathcal{H})) \cdot \left[\sum_{i \in I} 2^i \log(n_i + 1) \; + \; \sum_{i \in I} 2^i \log N_i^0 \right]. \tag{7}$$

Expanding the second summation we have (letting $\ell := \log\log m$ to simplify the notation)

$$\sum_{i \in I} 2^i \log N_i^0 \leq \sum_{i=0}^{\ell} 2^{i+1} \log\left(10(\ell - i + 1)^2\right) = 2^{\ell+2} \sum_{i=0}^{\ell} \frac{\log\left(10(\ell - i + 1)^2\right)}{2^{\ell - i + 1}} \leq O(2^{\ell+2}) = O(\log m),$$

where the last inequality follows form the fact that the series $\sum_{x=1}^{\infty} \frac{\log(10x^2)}{2^x}$ converges. Using this observation on (7) we get

$$\#\text{examples sent under } E \;=\; O(\mathcal{TS}(\mathcal{H})) \cdot \left(\log m + \sum_{i \in I} 2^i \log(n_i + 1) \right).$$

This concludes the proof of Theorem 3. □

C Teaching in the non-realizable case

In this section we provide the Teacher's algorithm for the case where the best hypothesis in $\mathcal{H}$ has $k > 0$ errors and prove Theorem 4. The algorithm makes use of an algorithm for *online fractional generalized set cover* [5], which is our starting point.

C.1 Online fractional generalized set cover

The offline version of this problem is that of finding an optimal solution to the following covering problem

$$\begin{aligned} \min \ & \sum_{e \in [m]} w_e \\ \text{s.t. } & \langle a^t, w \rangle \geq b_t \quad \forall t = 1, \dots, \ell \\ & w \in [0,1]^m, \end{aligned} \tag{8}$$

where the vectors a^t have 0/1 entries. This can be interpreted as follows: There is a ground set of ℓ elements and m sets (a^t_i is 1 iff the set i contains the element t), and the goal is to find a small fractional selection w of the sets so as to cover each element t at least b_t times.

In the online version of the problem the constraints $\langle a^1, w \rangle \geq b_1$, $\langle a^2, w \rangle \geq b_2$, ..., $\langle a^\ell, w \rangle \geq b_\ell$ are only revealed one-by-one, and the algorithm needs to maintain a sequence of feasible fractional solutions $w^1 \leq w^2 \leq \ldots \leq w^\ell$ that is pointwise non-decreasing (i.e., the algorithm cannot de-select items). More precisely, in the beginning of the game the algorithm has no information, and at round t the adversary reveals the constraint $\langle a^t, w \rangle \geq b_t$. Using only information seen thus far, the algorithm need to find a solution $w^t \in [0,1]^m$ that is pointwise $w^t \geq w^{t-1}$ and satisfies this new constraint (monotonicity guarantees that the previous constraints are also satisfied). The goal is to minimize the size of the solution in the last round, namely $\sum_e w^\ell_e$.

Several variations and generalizations of this problem have been studied, but for us it suffices that [5] gave an $O(\log m)$-approximation for this problem, namely the cost $\sum_e \bar{w}^\ell_e$ of the returned sequence $(\bar{w}^t)_{t \in [\ell]}$ is always at most $O(\log m)$ times the cost of the optimal solution w^* for the offline problem (8).

C.2 Algorithm and analysis

The main idea behind the algorithm is the equivalence between k-extended teaching sets and integral solutions to (8). To see this, recall that k is the minimum number of errors of a hypothesis in $\mathcal{H}$, and let $\mathcal{H}_{bad} \subseteq \mathcal{H}$ be the hypotheses with more than k errors. Recall that selection $\mathcal{X}'$ of examples is a k-extended teaching set if for all $h \in \mathcal{H}_{bad}$ the set $\mathcal{X}'$ contains at least $k+1$ examples where h is wrong.

But if we let $a(h) \in \{0,1\}^{\mathcal{X}}$ be the indicator vector of the examples where hypothesis h is wrong, then k-extended teaching sets correspond precisely to 0/1 solutions $w \in \{0,1\}^m$ of the problem (8) with constraints $\langle a(h), w \rangle \geq k+1$ for all $h \in \mathcal{H}_{bad}$.

Since the Teacher does not known the hypotheses $\mathcal{H}$, the idea is to:

1. Construct the online version of the instance by adding the constraints $\langle a(H_t), w \rangle \geq k+1$ one-by-one as hypotheses are received from the Learner

2. Use the algorithm from [5] to compute the fractional selections $W^1, W^2, \ldots \in [0,1]^{\mathcal{X}}$ of examples for each time step

3. Use W^t as a weighting to sample the examples, which are then sent to the Learner.

The additional element is that the minimum number of errors k is not known a priori. To handle this, the algorithm keeps a lower bound κ of k, initialized at 0. By definition our Learner always returns a hypothesis that makes the smallest possible number of errors over the examples sent thus far. Hence, if it sends a hypothesis H_t with err errors on the examples sent thus far, then we know that err is also a lower bound on k, and is used to update κ.

The final algorithm is presented in Figure 5. (We will use $\widetilde{W}^t \in \{0,1\}^m$ to denote the incidence vector of the examples sent by the end of time t, an hence $\langle a(h), \widetilde{W}^t \rangle$ is the number of examples sent thus far where h is wrong.)

```
**Input:** Examples $\mathcal{X}$, number of Learner's hypotheses $n = |\mathcal{H}|$, estimate $N$ of $n$
Initialize lower bound $\kappa = 0$ on $k$ (and let $W^0 = 0$)
For each time step $t = 1, 2, \ldots$
1. Receive hypothesis $H_t$ from Learner
2. **(Lower bound update)** Set $\kappa$ to be the maximum between its current value and the number
   of examples sent thus far where $H_t$ is wrong, i.e. $\kappa \leftarrow \max\{\kappa, \langle a(H_t), \widetilde{W}^{t-1} \rangle\}$
3. If the total number of errors of $H_t$ is $\kappa$ (which is $\leq k$), **return** $H_t$
4. **(Weight update)** Send constraint $\langle a(H_t), w \rangle \geq \kappa + 1$ to the fractional generalized set cover
   algorithm of [5], and receive from it an updated fractional solution $W^t$
5. **(Sampling)** Let $D_e^t = W_e^t - W_e^{t-1}$. For each example $e$, flip a coin with bias $\min\{D_e^t \cdot$
   $10\log(Nm), 1\}$, and send this example to Learner in case of heads
6. Let $\widetilde{W}^t = \{0,1\}^{\mathcal{X}}$ be the indicator vector of the examples sent thus far
7. If $\langle a(H_t), \widetilde{W}^t \rangle < \kappa + 1$, **return** FAIL
```

Figure 5: Teacher's algorithm for teaching a non-realizable hypothesis

Main analysis. We start collecting some simple observations about the algorithm.

Lemma 7. *Regardless of the estimate N, the algorithm from Figure 5 satisfies:*

1. *κ is always at most k*
2. *If the algorithm returns in Step 3, it returns a classifier with the minimum number of total errors k*
3. *The algorithm terminates in at most $(k+1)n + 1$ time steps.*

Proof. The first item was already discussed above, and the second item follows directly from the first.

To prove the third item, call a *phase* the sequence of time steps that had the same κ after the update in Step 2. Since κ increases by at least 1 when we start a new phase there are at most $k+1$ phases. To get the desired upper bound on the number of time steps, it then suffices to argue that if a hypothesis h appears twice in the same phase, then the algorithm returns in this second appearance. For that, consider a time t in a phase $\bar{\kappa}$. Because of the κ update step, in the beginning of time t the hypothesis H_t is wrong in at most $\bar{\kappa}$ examples sent thus far, namely $\langle a(H_t), \widetilde{W}^{t-1} \rangle \leq \bar{\kappa}$. However, by the end of of time t, after sending the new examples, either this number increases to at least $\bar{\kappa} + 1$ (i.e., $\langle a(H_t), \widetilde{W}^t \rangle \geq \bar{\kappa} + 1$), or the algorithm exits. Either way, this means that this same hypothesis H_t cannot appear in a later time step in the same phase. This concludes the proof. □

To conclude the analysis we need to show that with high probability the algorithm indeed returns in Step 3 and also sends a small number of examples. This requires the rather technical concentration inequality from Lemma 10 that handles all the correlations in the process. To improve readability we postpone the statement and proof of this lemma to Section C.3.

Lemma 8. *Regardless of the estimate N, with probability at least $1 - \frac{1}{2m^2 N}$ the number of examples sent by the algorithm is $O(\mathcal{TS}_k \log m \cdot \log Nm)$.*

Proof. Let τ be a random variable denoting the last time step of the algorithm. First we claim that $\sum_e W_e^\tau \le O(\mathcal{TS}_k \log m)$, that is, the total number of examples that are fractionally selected is at most $O(\mathcal{TS}_k \log m)$. To see this, fix any scenario and notice that the generalized fractional set cover instance sent to the algorithm of [5] is

$$\begin{aligned} \min \;& \sum_{e \in \mathcal{X}} w_e \\ \text{s.t. } & \langle a(H_t), w \rangle \ge \kappa_t + 1 \quad \forall t = 1, \dots, \tau \\ & w \in [0,1]^{\mathcal{X}}, \end{aligned} \tag{9}$$

where κ_t denotes the value of κ in Step 4 at time t. Since from the previous lemma all κ_t's are at most k, we see that any k-extended teaching set gives an (integral) feasible solution for this problem, and thus its optimal value is at most $\mathcal{TS}_k$. The guarantee of the algorithm from [5] then gives that its solution satisfies

$$\sum_e W_e^\tau \le [\text{optimal value of (9)}] \cdot O(\log m) \le O(\mathcal{TS}_k \log m),$$

proving the claim.

Now we upper bound the number of examples sent by the algorithm, namely $\sum_e \widetilde{W}_e^\tau$. Let $\mathcal{F}_t$ be the history of the algorithm up to time t; more formally, $\mathcal{F}$ is the σ-algebra generated by $H_1, \dots, H_{t+1}$ and $\widetilde{W}^1, \dots, \widetilde{W}^t$.[3] Let $\widetilde{D}_e^t := \widetilde{W}_e^t - \widetilde{W}_e^{t-1}$ be the indicator of the examples sent at time t, so $\sum_e \widetilde{D}_e^t$ counts the number of examples sent at time t. The total "conditionally expected" number of examples sent by the algorithm is

$$\begin{aligned} \sum_{t \le \tau} \sum_e \mathbb{E}[\widetilde{D}_e^t \mid \mathcal{F}_{t-1}] &\le (10 \log Nm) \sum_{t \le \tau} \sum_e D_e^t && \text{(by sampling Step 5)} \\ &\le (10 \log Nm) \sum_e W_e^\tau && \text{(by definition of } D_e^t\text{)} \\ &\le O(\mathcal{TS}_k \log m \cdot \log Nm). && \text{(by the previous claim)} \end{aligned}$$

Then by concentration of measure, we should have that the actual number of examples sent $\sum_e \widetilde{W}_e^\tau = \sum_{t \le \tau} \sum_e \widetilde{D}_e^t$ is close to this conditional expectation. Since there are correlations between the vectors $\widetilde{D}^t$'s (and also the stopping time τ) we cannot directly apply standard concentration inequalities. But employing part 1 of Lemma 10 we have (letting $\beta := cst \cdot \mathcal{TS}_k \log m \log Nm$ for a sufficiently large constant cst)

$$\Pr\left(\sum_e \widetilde{W}_e^\tau \ge 2\beta\right) = \Pr\left(\sum_{t \le \tau} \sum_e \widetilde{D}_e^t \ge 2\beta \text{ and } \sum_{t \le \tau} \sum_e \mathbb{E}[\widetilde{D}_e^t \mid \mathcal{F}_{t-1}] \le \beta\right) \le e^{-\frac{3}{14}\beta} \le \frac{1}{2m^2 N}.$$

This upper bounds the number of examples sent by the algorithm as desired. □

[3] The last term H_{t+1} is needed only when Learner is not a deterministic function of the examples it receives.

Lemma 9. *If $N \geq \max\{n, 4\}$, the algorithm returns* Fail *with probability at most $\frac{1}{2m}$.*

Proof. As defined in the previous lemma, let τ be the last time step of the algorithm, κ_τ be the value of κ at this time, and $\mathcal{F}_t$ be the σ-algebra generated by the history $H_1, \ldots, H_{t+1}$ and $\widetilde{W}^1, \ldots, \widetilde{W}^t$. Recall that $\langle a(h), W^t \rangle = \sum_{e \in \text{wrong}(h)} W_e^t$ is the fractional mass of the examples by time t where hypothesis h is wrong, and $\langle a(h), \widetilde{W}^t \rangle = \sum_{e \in \text{wrong}(h)} \widetilde{W}_e^t$ is the number of examples actually sent by time t where h is wrong.

When the algorithm returns Fails we have $\sum_{e \in \text{wrong}(H_\tau)} \widetilde{W}_e^\tau \leq \kappa_\tau$ (a too-small number of wrong examples sent) but by Step 4 we always have $\sum_{e \in \text{wrong}(H_\tau)} W_e^\tau \geq \kappa_\tau + 1$ (a good mass of wrong examples); thus, it suffices to show that

$$\Pr\left(\sum_{e \in \text{wrong}(H_\tau)} \widetilde{W}_e^\tau \leq \kappa_\tau \quad \text{and} \quad \sum_{e \in \text{wrong}(H_\tau)} W_e^\tau \geq \kappa_\tau + 1 \right) \overset{\text{want}}{\leq} \frac{1}{2m}.$$

By taking a union bound over all possible values $h \in \mathcal{H}$ for H_τ, it suffices to prove that for each h

$$\Pr\left(\sum_{e \in \text{wrong}(h)} \widetilde{W}_e^\tau \leq \kappa_\tau \quad \text{and} \quad \sum_{e \in \text{wrong}(h)} W_e^\tau \geq \kappa_\tau + 1 \right) \overset{\text{want}}{\leq} \frac{1}{2mn}. \tag{10}$$

To prove this bound again the idea is to prove a similar inequality for each fixed value $L \in \{0, \ldots, k\}$ of κ_τ using a concentration inequality and then to take a union bound over all L's; as in the previous lemma, for the concentration we will use the decomposition into increments $W_e^\tau = \sum_{t \leq \tau} D_e^t$, and similarly for $\widetilde{W}_e^\tau$. Unfortunately, because of the truncation $\min\{\cdot, 1\}$ in the sampling step we need to work with RVs Z_e^t and $\widetilde{Z}_e^t$ that are modified versions of D_e^t and $\widetilde{D}_e^t$.

So for each example $e \in \text{wrong}(h)$, let τ_e be the first time when truncation happens in the sampling step for e, namely it is the first time t where $D_e^t > \frac{1}{10 \log Nm}$ (and set $\tau_e = \tau$ when no truncation occurs). Then Z_e^t is the sequence $D_e^1, D_e^2, \ldots$ stopped right before τ_e, namely $Z_e^t = D_e^t \cdot \mathbf{1}(t < \tau_e)$. We define $\widetilde{Z}_e^t$ in the same way but with respect to $\widetilde{D}_e^t$.

The next claim relates these variables with the event of interest (10).

Claim 1. *For all $e \in wrong(h)$ we have:*

1. $\sum_t \widetilde{Z}_e^t \leq \sum_{t \leq \tau} \widetilde{D}_e^t - 1$
2. $\sum_t Z_e^t \geq \sum_{t \leq \tau} D_e^t - 1$
3. *For all t,* $\mathbb{E}[\widetilde{Z}_e^t \mid \mathcal{F}_{t-1}] = (10 \log Nm)\, Z_e^t$.

In particular, the left-hand side of (10) is upper bounded by

$$\Pr\left[\sum_{e \in wrong(h)} \sum_t \widetilde{Z}_e^t \leq \kappa_\tau - |wrong(h)| \quad and \quad \sum_{e \in wrong(h)} \sum_t \mathbb{E}[\widetilde{Z}_e^t \mid \mathcal{F}_{t-1}] \geq (10 \log Nm)\big(\kappa_\tau - |wrong(h)| + 1\big) \right].$$

Proof. By the construction of the sampling scheme we have that when truncation happens the example is sent, namely $\widetilde{D}_e^{\tau_e} = 1$, and therefore,

$$\sum_{t \leq \tau} \widetilde{D}_e^t \geq \sum_{t < \tau_e} \widetilde{D}_e^t + 1 = \sum_{t < \tau_e} \widetilde{Z}_e^t + 1,$$

proving the first part of the claim. For the second part, since the total weight W_e^τ is at most 1,

$$\sum_{t \le \tau} D_e^t \le \sum_{t < \tau_e} D_e^t + 1 = \sum_{t < \tau_e} Z_e^t + 1.$$

For the third part, since the event $\mathbf{1}(t < \tau_e)$ is determined by the history $\mathcal{F}_{t-1}$, conditioning on such history we have

$$\begin{aligned} \mathbb{E}[\widetilde{Z}_e^t \mid \mathcal{F}_{t-1}] &= \mathbf{1}(t < \tau_e) \cdot \mathbb{E}[\widetilde{D}_e^t \mid \mathcal{F}_{t-1}] \\ &= \mathbf{1}(t < \tau_e)(10 \log Nm) D_e^t \\ &= (10 \log Nm) Z_e^t, \end{aligned}$$

where the second equality holds because given any history if truncation happen at time t both sides equal 0, and if it does not happen then by the sampling procedure $\mathbb{E}[\widetilde{D}_e^t \mid \mathcal{F}_{t-1}] = (10 \log Nm) D_e^t$. This concludes the proof. □

Now for an integer L define the event

$$E_L := \left[\sum_{e \in \text{wrong}(h)} \sum_t \widetilde{Z}_e^t \le L \;\; \text{and} \quad \sum_{e \in \text{wrong}(h)} \sum_t \mathbb{E}[\widetilde{Z}_e^t \mid \mathcal{F}_{t-1}] \ge (10 \log Nm)\big(L+1\big) \right].$$

For $L \ge 0$, part 2 of the concentration inequality Lemma 10 shows that $\Pr(E_L) \le \frac{1}{4Nm^2}$. Since the event never holds for $L < 0$ due to the non-negativity of the $\widetilde{Z}_e^t$'s, taking a union bound we get that the probability at the end of Claim 1 can be upper bounded by $\sum_{L \in \{-n, \ldots, k\}} \Pr(E_L) \le \frac{k+1}{4Nm^2} \le \frac{1}{2mn}$, where the last inequality uses fact $k + 1 \le 2k \le 2m$ and the assumption $N \ge n$. This proves inequality (10), and concludes the proof of the lemma. □

Wrapping up. If then number of hypotheses n is known (and at least 4), we can run the algorithm from Figure 5 with $N = n$ and take a union bound over Lemmas 8 and 9 to obtain that with probability at least $1 - \frac{1}{m}$ the algorithm returns a minimum error hypothesis and sends at most $O(\mathcal{TS}_k \log m \cdot \log mn)$ examples.

To remove the assumption that n is known, run the algorithm with increasing estimates $N = N^i$ with $N^i := \frac{1}{m} 2^{2^i}$ for $i = \lceil \log \log 4m \rceil, \ldots, \lceil \log \log nm \rceil$. The probability that any of these runs sends more examples then the bound from Lemma 8 is at most $\frac{1}{2m} \sum_{i \ge 1} \frac{1}{2^{2^i}} \le \frac{1}{2m}$. Thus, with probability at least $1 - \frac{1}{2m}$ these runs send in total at most

$$\sum_{i=1}^{\lceil \log \log nm \rceil} O(\mathcal{TS}_k \log m \cdot \log mN_i) = O(\mathcal{TS}_k \log m) \sum_{i=1}^{\lceil \log \log nm \rceil} 2^i \le O(\mathcal{TS}_k \log m \cdot \log mn)$$

examples. Moreover, from Lemma 9 the last of these runs does not fail (so returns a minimum error hypothesis) with probability at least $1 - \frac{1}{2m}$. Taking a union bound over these guarantees then concludes the proof of Theorem 4.

C.3 Technical lemma

The following concentration inequality can be considered as an extension of Bernstein's inequality to the martingale case that is suitable for our use.

Lemma 10. *Consider a filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots$, and an adapted sequence of random vectors $X^1, \ldots, X^T \in \{0,1\}^m$ that may be correlated but satisfy the following: conditioned on $\mathcal{F}_{t-1}$, the coordinates of X^t are independent: for all $x \in \{0,1\}^n$*

$$\Pr(X^t = x \mid \mathcal{F}_{t-1}) = \prod_i \Pr(X^t_i = x_i \mid \mathcal{F}_{t-1}).$$

Let $Y^t = \sum_i X^t_i$. Then for any stopping time τ adapted to $\{\mathcal{F}_t\}_t$, $\alpha > 0$, and $\lambda \geq \alpha + 1$

$$\Pr\left(\sum_{t \leq \tau} Y^t \geq \alpha + 2\lambda \text{ and } \sum_{t \leq \tau} \mathbb{E}[Y^t \mid \mathcal{F}_{t-1}] \leq \alpha + \lambda\right) \leq e^{-\frac{3}{14}\lambda},$$

and

$$\Pr\left(\sum_{t \leq \tau} Y^t \leq \alpha \text{ and } \sum_{t \leq \tau} \mathbb{E}[Y^t \mid \mathcal{F}_{t-1}] \geq \alpha + \lambda\right) \leq e^{-\frac{3}{14}\lambda}.$$

We will need Freedman's Inequality.

Lemma 11 (Freedman's Inequality [9])**.** *Consider a filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots$ and an adapted martingale difference sequence $M_1, \ldots, M_T$ taking values in $[-1,1]$. Let $V = \sum_t \mathbb{E}[M_t^2 \mid \mathcal{F}_{t-1}]$ denote the predictable variance. Then for all $\lambda, v > 0$*

$$\Pr\left(\sum_t M_t \leq -\lambda \text{ and } V \leq v\right) \leq \exp\left(-\frac{\lambda^2}{2(v + \lambda/3)}\right),$$

and the same upper bound holds for the event $[\sum_t M_t \geq \lambda \text{ and } V \leq v]$.

Proof of Lemma 10. We first prove the lemma for $m = 1$, in which case we have $Y^t = X^t_1$. To simplify the notation let $\mathbb{E}_{t-1}[\cdot]$ denote the conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$ ($\mathbb{E}_0[\cdot] := \mathbb{E}[\cdot]$). The idea is to apply Freedman's Inequality to the sequence $Y^t - \mathbb{E}_{t-1}Y^t$, but to control the predictable variation we need to further stop the process.

Let τ' be the smallest t such that $\mathbb{E}_0 Y^1 + \ldots + \mathbb{E}_{t-1} Y^t \geq \alpha + \lambda$ (and $\tau' = \infty$ if there is no such t). Define the truncated process $\overline{Y}^t := Y^t \cdot \mathbf{1}(t \leq \tau) \cdot \mathbf{1}(t \leq \tau')$. We claim that this truncated process upper bounds the original one, namely

$$\Pr\left(\sum_{t \leq \tau} Y^t \geq \alpha + 2\lambda \text{ and } \sum_{t \leq \tau} \mathbb{E}_{t-1} Y^t \leq \alpha + \lambda\right) \leq \Pr\left(\sum_t \overline{Y}^t \geq \alpha + 2\lambda \text{ and } \sum_t \mathbb{E}_{t-1}\overline{Y}^t \leq \alpha + \lambda\right) \tag{11}$$

$$\Pr\left(\sum_{t \leq \tau} Y^t \leq \alpha \text{ and } \sum_{t \leq \tau} \mathbb{E}_{t-1} Y^t \geq \alpha + \lambda\right) \leq \Pr\left(\sum_t \overline{Y}^t \leq \alpha \text{ and } \sum_t \mathbb{E}_{t-1}\overline{Y}^t \geq \alpha + \lambda\right). \tag{12}$$

To justify the first inequality, notice that for every scenario satisfying the LHS the truncation τ' does not kick in, and hence the scenario satisfies the RHS. For the second inequality, since $\sum_t \overline{Y}^t \leq \sum_{t \leq \tau} Y^t$, for every scenario satisfying the LHS we have $\sum_t \overline{Y}^t \leq \alpha$; moreover, for every scenario in the LHS, we have $\sum_t \mathbb{E}_{t-1}\overline{Y}^t \geq \alpha + \lambda$: since in this case $\sum_{t \leq \tau} \mathbb{E}_{t-1} Y^t \geq \alpha + \lambda$ we have $\tau' \leq \tau < \infty$ and hence $\sum_t \overline{Y}^t = \sum_{t \leq \tau'} \overline{Y}^t \geq \alpha + \lambda$, the last inequality by definition of τ'. So every scenario that belongs to the LHS also belongs to the RHS, thus proving (12).

Now define the martingale difference sequence $M_t := \overline{Y}^t - \mathbb{E}_{t-1}\overline{Y}^t$ and notice that

$$\Pr\left(\sum_t \overline{Y}^t \geq \alpha + 2\lambda \text{ and } \sum_t \mathbb{E}_{t-1}\overline{Y}^t \leq \alpha + \lambda\right) \leq \Pr\left(\sum_t M_t \geq \lambda\right) \tag{13}$$

$$\Pr\left(\sum_t \overline{Y}^t \leq \alpha \text{ and } \sum_t \mathbb{E}_{t-1}\overline{Y}^t \geq \alpha + \lambda\right) \leq \Pr\left(\sum_t M_t \leq -\lambda\right). \tag{14}$$

Moreover, because of the truncation we can bound the predictable variance of this martingale: in our current case $m = 1$ we have $|\overline{Y}^t| \leq 1$ and hence

$$\mathbb{E}_{t-1}M_t^2 = \operatorname{Var}(\overline{Y}^t \mid \mathcal{F}_{t-1}) \leq \mathbb{E}_{t-1}(\overline{Y}^t)^2 \leq \mathbf{1}(t \leq \tau') \cdot \mathbb{E}_{t-1}Y^t.$$

So using the definition of the stopping time τ' and the fact $\mathbb{E}_{t-1}Y^t \leq 1$, the predictable variance is $V := \sum_t \mathbb{E}_{t-1}M_t^2 \leq \sum_{t \leq \tau'} \mathbb{E}_{t-1}Y^t \leq \alpha + \lambda + 1$.

Finally, applying Lemma 11 to $(M_t)_t$ with $v = \alpha + \lambda + 1$ we get

$$\Pr\left(\sum_t M_t \geq \lambda \text{ and } V \geq \alpha + \lambda + 1\right) \leq \exp\left(-\frac{\lambda^2}{2(\alpha + \lambda + 1 + \lambda/3)}\right) \leq e^{-\frac{3}{14}\lambda},$$

where last inequality uses the assumption $\lambda \geq \alpha + 1$. The same bound also holds for $\sum_t M_t \leq -\lambda$. Chaining these bounds with inequalities (11)-(14). This concludes the proof for the case $m = 1$.

For the case $m > 1$ we simply reveal the random variables $X_1^t, X_2^t, \ldots$ one-by-one and apply the result for the case $m = 1$ to this sequence. More precisely, we index time using (t, i) and the lexicographic order $(1,1), \ldots, (1,n), (2,1), \ldots$, and use $prec(t,i)$ to denote the predecessor of (t,i), and define $\widetilde{Y}^{t,i} := X_i^t$. Let $\widetilde{\mathcal{F}}_{t,m} := \mathcal{F}_t$, and for $i < m$ let $\widetilde{\mathcal{F}}_{t,i}$ be the σ-algebra generated by $\mathcal{F}_{t-1}$ and $X_1^t, \ldots, X_i^t$. The main properties of this filtration are: 1) $\widetilde{Y}^{t,i}$ is $\widetilde{\mathcal{F}}_{t,i}$-measurable; 2) by the conditional independence of the RV's $(X_1^t, \ldots, X_n^t)|_{\mathcal{F}_{t-1}}$, for $i < m$ conditioning on $\widetilde{\mathcal{F}}_{prec(t,i)}$ is the same as conditioning on $\mathcal{F}_{t-1}$, or more precisely

$$\mathbb{E}[\widetilde{Y}^{t,i} \mid \widetilde{\mathcal{F}}_{prec(t,i)}] = \mathbb{E}[\widetilde{Y}^{t,i} \mid \mathcal{F}_{t-1}] = \mathbb{E}_{t-1}X_i^t. \tag{15}$$

Notice that $\sum_{(t,i) \preceq (\tau,m)} \widetilde{Y}^{t,i} = \sum_{t \leq \tau} \sum_{i \leq m} X_i^t = \sum_{t \leq \tau} Y^t$, and using (15) we have $\sum_{(t,i) \preceq (\tau,m)} \mathbb{E}[\widetilde{Y}^{t,i} \mid \widetilde{\mathcal{F}}_{prec(t,i)}] = \sum_{t \leq \tau} \sum_{i \leq m} \mathbb{E}_{t-1}X_i^t = \sum_{t \leq \tau} \mathbb{E}_{t-1}Y^t$. So applying the current lemma for the case $m = 1$ to $(\widetilde{Y}^{t,i})_{t,i}$ and the stopping time (τ, m) proves the full statement of the lemma. This concludes the proof. □

D Inapproximability result on randomized algorithms for teaching a black box learner

In this section we rephrase the lower bound on the competitiveness of any polynomial time algorithm for the on line set cover from [14] in terms of approximation of the teaching set for a randomized teacher dealing with a black box learner. This bound together with the result in section 3.1 shows a separation between the efficiency of teaching achievable with an adversarial learner as compared to the teaching efficiency achievable with a smooth transition learner.

Let $\mathcal{I} = \{a_1, \ldots, a_N\}, \mathcal{S} = \{S_1, \ldots, S_M\}$ be the set of items and the family of sets in an (off line) set cover instance, i.e., $S_i \subseteq \mathcal{I}$ for each $i = 1, \ldots, M$.

For $i = 1, \ldots, N-1$, let $\mathcal{I}^{(i)}, \mathcal{S}^{(i)}$ be distinct copies of $\mathcal{I}$ and $\mathcal{S}$, i.e., we have that

- for each $i \neq i'$, it holds that $\mathcal{I}^{(i)} \cap \mathcal{I}^{(i')} = \emptyset$;
- for each i, it holds that $\mathcal{S}^{(i)} \in 2^{\mathcal{I}^{(i)}}$;
- for each i, ℓ, t, it holds that $a_\ell^{(i)} \in S_t^{(i)}$ iff $a_\ell \in S_t$;

hence, in particular, for each i, ℓ, j, t it holds that $a_\ell^{(i)} \in S_t^{(j)}$ iff $i = j$ and $a_\ell \in S_t$.

In the following, w.l.o.g, let us assume that N is a power of 2.

We let

$$\mathcal{X} = \{E_\ell^{(j)} \mid j = \frac{N}{2}, \ldots, N-1, \ \ell = 1, \ldots, M\} \cup \{\tilde{E}\}$$

be the set of examples of our instance, where $\tilde{E}$ is a special example whose role will be clarified shortly.

For each $i = 1, \ldots, N-1$ and $t = 1, \ldots, N$ we define a hypothesis $h_t^{(i)} : \mathcal{X} \mapsto \{0, 1\}$ by setting $h_t^{(i)}(\tilde{E}) = 0$ and $h_t^{(i)}(E_\ell^{(j)}) = 1$ if and only if $a_t \in S_\ell$ and $i \in \{j, \lfloor j/2 \rfloor, \ldots, 1\}$.

We then define two additional hypotheses $h^* : \mathcal{X} \mapsto \{0, 1\}$ and $\tilde{h} : \mathcal{X} \mapsto \{0, 1\}$ by

$$\begin{aligned} h^*(E) &= 0, \text{ for each } E \in \mathcal{X}, \\ \tilde{h}(E) &= 1, \text{ iff } E = \tilde{E}. \end{aligned}$$

Note that $\tilde{h}$ differs from h^* in only one example, namely $\tilde{E}$ is the only example that *covers* $\tilde{h}$. Equivalently, with respect to the analysis of the smooth transition learner, $\tilde{h}$ is at distance 1 from h^*.

Now we define $N/2$ classes of hypotheses. For that, consider a binary tree with $N-1$ nodes, which are labelled $1, \ldots, N-1$, proceeding from the root to the leaves and from left to right on each level, so that the root is the node 1, its left child is the node 2, the right child is the node 3, and so on, with the leaves, that from left to right are the nodes $\frac{N}{2}, \frac{N}{2}+1, \ldots, N-1$. Think of each node j as containing the sequence of hypotheses $(h_1^{(j)}, \ldots, h_N^{(j)})$. The class of hypotheses $\mathcal{H}_i$, for $i = N/2, \ldots, N-1$ contains the hypotheses h^* and $\tilde{h}$ and also every hypothesis that belongs to some node (sequence) in the path from the root to the leaf i.

Let $\mathcal{U} = \cup_i \mathcal{H}_i$. The machine teaching instance is defined by the set of examples $\mathcal{X}$ and the universe of hypotheses $\mathcal{U}$, where, unknowingly to the teacher, the learner's hypothesis class lies. In particular, the hypothesis class of the learner will be one of the sets $\mathcal{H}_i$.

To explain the behaviour of the Learner, we need to define a sequence ρ_i associated to the class $\mathcal{H}_i$. This sequence starts with $\tilde{h}$ and then is obtained by concatenating the sequences in the path from the root to the leaf i of the tree. Finally, h^* is appended to sequence ρ_i.

Consider now a Learner (being the adversary) that with uniform probability chooses $i \in \{N/2, N/2+1, \ldots, N-1\}$—equivalently has the hypothesis class $\mathcal{H}_i$—and gives to the teaching algorithm consistent hypotheses following the sequence ρ_i, skipping those made inconsistent by the received examples. Note that, in any case, the first hypothesis presented will be $\tilde{h}$ and any teaching algorithm must cover it with $\tilde{E}$. Therefore, we can assume that the first interactions are $\tilde{h}, \tilde{E}, h_1^{(1)}$.

The key point here is that every teaching algorithm must use $\tilde{E}$ and then behaves exactly as if the instance did not contain $\tilde{h}$ and $\tilde{E}$. This latter instance is the one that can be proved to have a lower bound of $\mathcal{TS} \log m \log n$ using the argument in [14]: (1) Every deterministic algorithm in expectation over the possible $N/2$ choices of the sequence ρ_i, will spend at least

$\frac{\log N}{2}k$ examples where k is equal to the size of the minimum size teaching set, i.e., the size of the optimal set cover for the off-line instance, because, without knowing the choice of ρ_i the example chosen by the algorithm to cover some new hypothesis has 1/2 probability of covering some hypothesis presented later; (2) A randomized algorithm, seen as a distribution over deterministic algorithms, on average over the $N/2$ possible input sequences, uses less than $\frac{\log N}{2}k$ examples with probability $< 2/3$. Leveraging these two observation and the gap reduction from SAT to Set Cover of [21] one obtains the following.

Theorem 10. *There exists a constant $d > 0$ such that the following holds: If there exists a polynomial time randomized algorithm for teaching a black box learner that guarantees to use less than $d \cdot \mathcal{TS} \log n \log m$ examples on every instance where the set of examples has size m, the learner's hypothesis class has size n then $NP \subset BPP$.*

This shows that even if Learner starts with the hypothesis $\tilde{h}$ close to the target h^*, there is no way for the teacher algorithm to build a teaching set of size $o(\mathcal{TS} \log m \log n)$.

E Random Learner model

E.1 Proof of Lemma 5

For each example $e \in \mathcal{X}$, let c_e (resp. $\tilde{c}_e$) be the number of hypotheses of $\mathcal{H}_t$ (resp. $\tilde{\mathcal{H}}_t$) covered by e.

Let e^* be an example that covers the largest number of hypotheses in $\mathcal{H}_t$ so that $c_{e^*} = c^*(\mathcal{H}_t)$. Thus, $Pr[c_{\tilde{e}} < \frac{1}{12}c^*(\mathcal{H}_t)]$ can be upper bounded by the probability that some example b, with $c_b < \frac{1}{12}c_{e^*}$, covers at least the same number of hypotheses in $\tilde{\mathcal{H}}_t$ covered by e^*, that is, $\tilde{c}_b \geq \tilde{c}_{e^*}$.

For a fixed example b, with $c_b < \frac{1}{12}c_{e^*}$, let E_b be the event in which $\tilde{c}_b \geq \tilde{c}_{e^*}$. It is enough to show that $Pr[E_b] \leq 2/m^5$ since we can take the union bound, considering all the examples, to show that the probability of E_b happens for some b is limited above by $2/m^4$.

Let z be a real number and let F^1 and F^2 be the events in which $\tilde{c}_{e^*} \leq z$ and $\tilde{c}_b \geq z$, respectively. We have that $E_b \subseteq F^1 \cup F^2$. In fact, if E_b happens and $\tilde{c}_{e^*} \leq z$ then F^1 happens. Otherwise, if E_b happens and $\tilde{c}_{e^*} > z$ then F^2 happens. Thus, $Pr[E_b] \leq Pr[F^1] + Pr[F^2]$ so that it suffices to obtain upper bounds on both $Pr[F^1]$ and $Pr[F^2]$. In what follows, we use $z = \frac{1}{2}\frac{k \cdot c_{e^*}}{|\mathcal{H}_t|}$, where $k \geq 40\,\mathcal{TS}(\mathcal{H}_t, h^*) \ln m$.

Let $\tilde{\mathcal{H}}_t = (h_1, \ldots, h_k)$ and for $i = 1, \ldots, k$, let X_i^e be a random variable defined as

$$X_i^e = \begin{cases} 1 & \text{if } e \in \text{wrong}(h_i) \\ 0 & \text{if } e \notin \text{wrong}(h_i). \end{cases}$$

Then, $\Pr(X_i^e = 1) = \frac{c_e}{|\mathcal{H}_t|}$. Let $X^e = \sum_{i=1}^k X_i^e = \tilde{c}_e$ and $\mu_e = k \cdot \frac{c_e}{|\mathcal{H}_t|} = \sum_{i=1}^k E[X_i^e] = E[X^e]$,

By Chernoff bound(see, e.g., [19, (4.5)]), we have that, for any $0 < \delta < 1$,

$$\Pr\left(\frac{\tilde{c}_e}{k} \leq (1-\delta)\frac{c_e}{|\mathcal{H}_t|}\right) = \Pr\left(X^e \leq (1-\delta)\mu_e\right) \leq e^{-\frac{\mu_e \delta^2}{2}} = e^{-\frac{c_e}{|\mathcal{H}_t|}\frac{k\delta^2}{2}} \tag{16}$$

From (16), with $\delta = 1/2$ and $e = e^*$, we get

$$Pr[F^1] = \Pr\left(\frac{\tilde{c}_{e^*}}{k} \leq \frac{1}{2}\frac{c_{e^*}}{|\mathcal{H}_t|}\right) \leq e^{-\frac{c_{e^*}}{|\mathcal{H}_t|}\frac{k}{8}}, \tag{17}$$

It is not hard to see that there must be an example that covers at least $|\mathcal{H}_t|/(\mathcal{TS}(\mathcal{H}_t, h^*))$ hypotheses in $\mathcal{H}_t$, hence, in particular, $c_{e^*} \geq \frac{|\mathcal{H}_t|}{\mathcal{TS}(\mathcal{H}_t, h^*)}$.

Because of $k \geq 40\,\mathcal{TS}(\mathcal{H}_t, h^*)\ln m$, it follows that

$$Pr[F^1] = \Pr\left(\frac{\tilde{c}_{e^*}}{k} \leq \frac{1}{2}\frac{c_{e^*}}{|\mathcal{H}_t|}\right) \leq \frac{1}{m^5}, \tag{18}$$

i.e., with high probability the example e^* will appear as one that covers a large fraction of hypotheses in $\tilde{\mathcal{H}_t}$, too.

To bound $Pr[F^2]$ we rely on the following Chernoff bound (see [19, (4.3)]). For any $R \geq 6\mu_e$, we have

$$\Pr(X^e > R) \leq 2^{-R}. \tag{19}$$

Under the assumption that $c_b < \frac{1}{12}c_{e^*}$ we have $\frac{1}{2}\mu_{e^*} > 6\mu_b$. Let $R = \frac{1}{2}\mu_{e^*}$. Then, by (19) we have

$$Pr[F^2] = \Pr\left(\frac{\tilde{c}_b}{k} \geq \frac{1}{2}\frac{c_{e^*}}{|\mathcal{H}_t|}\right) \leq 2^{-\frac{k}{2}\frac{c_{e^*}}{|\mathcal{H}_t|}} \leq \frac{1}{m^5}, \tag{20}$$

where in the last inequality we used the assumption $k \geq 10\mathcal{TS}(\mathcal{H}_t, h^*)\log_2 m$. This conclude the proof.

E.2 Proof of Lemma 6

Let $C(\mathcal{H})$ be the total number of examples selected by the algorithm on an initial set of hypotheses $\mathcal{H}$, of cardinality n.

We can show by induction on the cardinality of $\mathcal{H}$ that $C(\mathcal{H}) \leq \frac{1}{\alpha}\ln(|\mathcal{H}|)$. The inequality holds true for $|\mathcal{H}| = 1$. We can focus on the induction step.

Let e be the first example sent by the algorithm and let c_e be the number of hypotheses that x covers in $\mathcal{H}$.

Let $\mathcal{H}_1$ be the set of consistent hypotheses after sending the example e. We have that $\mathcal{H}_1$ has cardinality $n - c_e \leq n - \alpha\, c^*(\mathcal{H})$, and we have that:

$$C(\mathcal{H}) \leq 1 + C(\mathcal{H}_1).$$

Since $\mathcal{H}_1 \subset \mathcal{H}$ and $c^*(\mathcal{H}) \geq \frac{|\mathcal{H}|}{\mathcal{TS}(\mathcal{H},h^*)}$, we have the following lower bound on $\mathcal{TS}(\mathcal{H}, h^*)$

$$\mathcal{TS}(\mathcal{H}, h^*) \geq \max\left\{\frac{|\mathcal{H}|}{c^*(\mathcal{H})}, \mathcal{TS}(\mathcal{H}_1, h^*)\right\}.$$

Therefore, putting together the last two inequality we have

$$\begin{aligned}\frac{C(\mathcal{H})}{\mathcal{TS}(\mathcal{H}, h^*)} &\leq \frac{c^*(\mathcal{H})}{|\mathcal{H}|} + \frac{C(\mathcal{H}_1)}{\mathcal{TS}(\mathcal{H}_1, h^*)} \leq \frac{1}{\alpha}\frac{c_e}{|H|} + \frac{C(\mathcal{H}_1)}{\mathcal{TS}(\mathcal{H}_1, h^*)} \\ &\leq \frac{1}{\alpha}\frac{c_e}{|\mathcal{H}|} + \frac{1}{\alpha}\ln(|\mathcal{H}_1|) = \frac{1}{\alpha}\frac{c_e}{n} + \frac{1}{\alpha}\ln(n - c_e) \leq \frac{1}{\alpha}\ln(n).\end{aligned}$$

This concludes the proof.

F Experiments on non-redundant teaching sets

To evaluate the impact of the algorithm from Section 4 we made some simulations using synthetic data. We measure the reduction on the number of examples that the method achieves over the teaching sets obtained when the Teacher is $\mathcal{A}_{\text{agno}}$ and the Learner always returns a random hypothesis consistent with examples received thus far.

In what follows, we describe the data employed in our experiments. We fixed the number of hypotheses $n = 30.001$ and varied the number of examples m in the set $\{100, 500, 2500, 12.500, 75.000\}$. We only use binary hypotheses, that is, $|\mathcal{Y}| = 2$. For every experiment the class of hypotheses $\mathcal{H}$ consists of a hypothesis h^* that makes no errors and also 5 groups of 6.000 hypotheses, where all the hypotheses of the i-th group fail in $(2i)\%$ of their examples. We focus on hypotheses that fail in a relatively small number of examples since those that make a lot of errors are easier to discard.

To generate these hypotheses we used a parameter $\alpha \in [0, 1]$ that defines the number of *difficult examples* and a parameter $\beta \in [0, 1]$ defining the number of errors associated with the difficult examples in each hypothesis. More specifically, to generate a class of hypotheses $\mathcal{H}(m, \alpha, \beta)$, we first fix the $m\alpha$ examples that are difficult. Then, the set of wrong examples for a hypothesis that fail in $x\%$ of its examples is built by randomly selecting $(x/100) \cdot m \cdot \beta\%$ examples from the difficult set and randomly selecting the remaining ones from the non-difficult set.

We consider 35 combinations of parameters (α, β, m)

$$\{\{0.1, 0.2, 0.3\} \times \{0.25, 0.5\} \times \{100, 500, 2500, 12.500, 75.000\}\} \cup \\ \{\{0\} \times \{0\} \times \{100, 500, 2500, 12.500, 75.000\}\}$$

For each combination we generated 10 classes of hypotheses and, then, we measured the reduction on the size of the teaching set obtained by $\mathcal{A}_{\text{agno}}$ when the algorithm from Section 4 is employed to remove redundant examples. On average, over the 350 runs, 38% of the examples are removed and we had cases in which this reduction was larger than 50%.

The code employed for this simulation is available in `https://github.com/sfilhofreitas/TeachingWithLimitedKnowledge`

G Computational Experiments - More details

We give more details on the experiments reported in Section 5 to compare the number of examples required by $\mathcal{A}_{\text{agno}}$ and `NIT` to attain a given level of accuracy.

Computational Environment. Our experiments were executed on an Intel Core i7-7700HQ with 16G RAM. We used Python 3.7.3 with the following libraries: numpy 1.16.4; pandas 0.23.4; scikit-learn 0.20.1 and lightgbm 2.3.1. The codes and the datasets are available in `https://github.com/sfilhofreitas/TeachingWithLimitedKnowledge`.

Datasets and Preprocessing. Our selected datasets meet the following criteria: they do not have missing values and have up to 60.000 labelled examples. The latter was used to prevent time consuming experiments.

Datasets were preprocessed to convert categorical attributes into numeric ones: if a categorical attribute takes k different values then it is replaced with k binary attributes. For that we used method `get_dumies` from pandas library. That was done because our learners do not handle categorical attributes directly.

Learner's implementations We use classes `sklearn.ensemble.RandomForestClassifier` and `lightgbm.LGBMClassifier`, with their default parameters, to implement Random forest and LGBM, respectively.

Teacher's implementations. Algorithm $\mathcal{A}_{\text{agno}}$ is explained and analysed in Section 2 for the setting in which the Learner always returns a hypothesis consistent with the examples sent thus far. However, in practice we do not have this guarantee. To run $\mathcal{A}_{\text{agno}}$ for a more general setting we use the following definition for the set $wrong(h)$:

$$wrong(h) = \{x \in \mathcal{X} \mid h \text{ fails on } x \text{ and } x \text{ has not been sent to } \mathsf{Learner}\}.$$

rather than the original one:

$$wrong(h) = \{x \in \mathcal{X} \mid h \text{ fails on } x\}.$$

With the additional requirement in the definition of $wrong(h)$ we avoid sending the same example more than once. Note that for the realizable case $h^* \in \mathcal{H}$, this new definition is equivalent to the original one since, in this case, the hypotheses received by the Teacher do not make errors on the examples sent thus far.

The other Teacher we consider, `NIT`, looks for a small random set of examples for which the Learner, when trained on this set, achieves accuracy larger than some given threshold over the full dataset under consideration. For that it picks a random order of the examples from the dataset and then it looks for the smallest value of ℓ for which the sample consisting of the first ℓ examples of this order satisfies the desired property. To avoid very expensive computations, the value of ℓ is increased using a step equal to 0.5% of the size of the dataset.

Results. To take into account the Teacher randomness, for each possible triple (Teacher, Learner, dataset) we had 30 runs. For a fixed combination (Learner,dataset), the first hypothesis available to both $\mathcal{A}_{\text{agno}}$ and `NIT`, for their ith run, is obtained by training the Learner with the first 1% examples of the random order employed by `NIT` for its ith run.

The Columns `Avg Size` of Tables 2 and 3 show the average number of examples (relative to the size of the dataset) required for each pair Teacher-Learner to obtain an accuracy over the full dataset larger than 99% of that achieved when the Learner is trained/tested in the full dataset. Table 2 presents results for Random forests while Table 3 present those for LGBM. As an example, for dataset `mnist`, the combination $\mathcal{A}_{\text{agno}}$-LGBM required on average (over the 30 runs) to be trained with ≈ 3,780 (6.3% of 60,000) labelled examples to attain accuracy with respect to the full dataset larger than 99% of that achieved when LGBM is trained and then tested over the 60,000 `mnist`'s examples. On the other hand, for the combination `NIT`-LGBM, $\approx 37,620$ (62.7 % of 60,000) examples are required on average. We note that the results presented in Table 1, for the target 99%, are the averages of the values shown on columns `Avg Size` of Tables 2 and 3.

We can observe a significant advantage of $\mathcal{A}_{\text{agno}}$: it has a standard deviation lower than `NIT` and, in terms of the number of examples required, it outperformed `NIT` in all combinations but (`credit_card`, LGBM). These results suggest that algorithm $\mathcal{A}_{\text{agno}}$ is potentially a very competitive option for fast training of classification methods.

Dataset	Size	$\mathcal{A}_{\text{agno}}$-Random Forest		NIT-Random Forest	
		Avg. Size (%)	StdDev (%)	Avg. Size (%)	StdDev (%)
avila	10,430	25.5	0.8	74.4	3.1
bank_marketing	41,188	22.4	0.3	89.2	0.7
car	1,728	24	1	77.7	4.8
credit_card	30,000	54	0.4	94	0.4
crowdsourced	10,545	19.5	0.5	82.4	1.8
Electrical_grid	10,000	1.1	0	2.1	0.6
Firm_Teacher	10,796	52.7	0.7	95.3	0.6
HTRU	17,898	7.1	0.2	46.6	2.5
mnist	60000	23.5	0.4	82.1	0.7
mushroom	8,124	1.2	0.1	4.4	1.7
nursery	12,960	13.2	0.5	59.1	2.6
Sensorless_drive_diagnosis	58,509	3	0.1	11.5	1.1
Average		20.6	0.4	59.9	1.7

Table 2: Column `Avg Size` shows the average number of examples (in percentage relative to the size of the dataset) required for each pair Teacher-Learner to obtain an accuracy over the full dataset larger than 99% of that achieved when the Learner (Random Forest) is trained/tested in the full dataset. Each value is the average of 30 runs.

Dataset	Size	$\mathcal{A}_{\text{agno}}$-LGBM		NIT-LGBM	
		Avg. Size (%)	StdDev (%)	Avg. Size (%)	StdDev (%)
avila	10,430	1.1	0	1.1	0
bank_marketing	41,188	12.7	1.4	27.1	2.3
car	1,728	14.3	0.6	47.1	4
credit_card	30,000	26.3	4.3	16.7	1.6
crowdsourced	10,545	8.7	0.2	74.8	2.4
Electrical_grid	10,000	1	0	1	0
Firm_Teacher	10,796	25.7	1.1	64.7	4.2
HTRU	17,898	5.5	0.3	19.5	3.2
mnist	60000	6.3	0.1	62.6	1.2
mushroom	8,124	1.1	0.1	4.5	1.4
nursery	12,960	2.7	0.2	9.7	1
Sensorless_drive_diagnosis	58,509	1.7	0.1	7.5	0.7
Average		8.9	0.7	28	1.8

Table 3: Column `Avg size` shows the average number of examples (in percentage relative to the size of the dataset) required for each pair Teacher-Learner to obtain an accuracy over the full dataset larger than 99% of that achieved when the Learner (LGBM) is trained/tested in the full dataset. Each value is the average of 30 runs.