A Resilient Distributed Boosting Algorithm

Yuval Filmus [*1] **Idan Mehalel** [*1] **Shay Moran** [*23]

Abstract

Given a learning task where the data is distributed among several parties, communication is one of the fundamental resources which the parties would like to minimize. We present a distributed boosting algorithm which is resilient to a limited amount of noise. Our algorithm is similar to classical boosting algorithms, although it is equipped with a new component, inspired by Impagliazzo's hard-core lemma (Impagliazzo, 1995), adding a robustness quality to the algorithm. We also complement this result by showing that resilience to any asymptotically larger noise is not achievable by a communication-efficient algorithm.

1. Introduction

Most work in learning theory focuses on designing efficient learning algorithms which generalize well. New considerations arise when the training data is spread among several parties: speech recorded on different smartphones, medical data gathered from several clinics, and so on. In such settings, it is important to minimize not only the computational complexity, but also the communication complexity. Apart from practical considerations of limited bandwidth, minimizing the communication complexity also limits the amount of data being exposed to prying ears. This motivates designing distributed learning algorithms, which improve on the naive idea of sending all training data to a single party.

In the classical PAC model, distributed learning has been studied mostly in the realizable setting, where it was shown that distributed implementations of boosting algorithms can learn any VC class with communication complexity which is polynomial in the description length (in bits) of a single example (Balcan, Blum, Fine, and Mansour, 2012; Daumé, Phillips, Saha, and Venkatasubramanian, 2012b; Kane, Livni, Moran, and Yehudayoff, 2019).

In this work we deviate from the realizable setting and allow a small amount of noise in the input sample. In our setting there are k players and a center, who are given a domain U of size $|U| = n$ and a concept class $\mathcal{H}$ over U with VC dimension $d \ll n$. For a labelled input sample S distributed among the players, let $OPT := OPT(S) \in \mathbb{N}$ denote the number of examples in S which are misclassified by the best hypothesis in $\mathcal{H}$. In most parts of the paper, we require that $OPT \in \mathsf{polylog}\, n$. The goal of the parties is to learn together a classifier f that has at most OPT errors on S, while using $\mathsf{poly}(d, k, \log|S|, \log n)$ bits of communication. Note that $\log n$ is the number of bits needed to encode a single point in U, and thus $\mathsf{polylog}\, n$ means polynomial in the description length of a single example.[1]

Main result. Our main result, formally stated in Theorems 2.2 and 2.3 asserts the following: for every VC class, if the minimal error of an hypothesis satisfies $OPT \in \mathsf{polylog}\, n$, then a simple robust variant of classical boosting learns it with $\mathsf{poly}(d, k, \log|S|, \log n)$ communication complexity. Conversely, when $OPT \notin \mathsf{polylog}\, n$, there exist one-dimensional VC classes for which *any* learning algorithm has super-polylogarithmic communication complexity.

The novelty of our algorithm lies in a non-standard usage of boosting that identifies small "hard" sets for which any hypothesis from the class has large error. This kind of usage resembles (and is inspired by) Impagliazzo's hard-core lemma (Impagliazzo, 1995), in particular its proof using the method of multiplicative weights. Our negative result is a slight extension of the argument appearing in (Kane, Livni, Moran, and Yehudayoff, 2019).

We note that our positive result can alternatively be obtained by a reduction to *semi-agnostic learning* (Bun, Kamath, Steinke, and Wu, 2019), that is, agreeing on a classifier with at most $c \cdot OPT$ errors for some constant c. Semi-agnostic learning is possible using $\mathsf{poly}(d, k, \log|S|, \log n)$ bits of communication by the works of Balcan, Blum,

[*]Equal contribution [1]The Henry and Marilyn Taub Faculty of Computer Science, Technion, Haifa, Israel [2]Faculty of Mathematics, Technion, Haifa, Israel [3]Google Research, Israel. Correspondence to: Idan Mehalel <idanmehalel@cs.technion.ac.il>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]We refer the reader to (Kane, Livni, Moran, and Yehudayoff, 2019; Braverman, Kol, Moran, and Saxena, 2019) for a more thorough discussion regarding the choice of $\mathsf{polylog}\, n$ as a "yardstick" for communication efficiency.

Fine, and Mansour (2012); Chen, Balcan, and Chau (2016). Given a semi-agnostic communication protocol with a constant approximation factor c and communication complexity $\mathsf{poly}(d, k, \log|S|, \log n)$, one can proceed as follows: execute the semi-agnostic protocol to obtain a hypothesis f, and have each player broadcast her examples that f misclassifies. Then, the players modify f on the misclassified points and output an optimal hypothesis f'. If there exists an hypothesis in the class whose error is $\mathsf{polylog}(n)$ then the communication cost of this step is $\mathsf{poly}(d, k, \log|S|, \log n)$, and thus the overall communication complexity is $\mathsf{poly}(d, k, \log|S|, \log n)$.

The advantage of our approach is the simplicity of our protocol, which is a simple modification of the classical boosting approach that makes it resilient to mild noise. This is in contrast with semi-agnostic learning protocol which rely on non-trivial subroutines (e.g. the distributed implementation of Bregman projection in the protocol by Chen, Balcan, and Chau (2016)).

Empirical loss versus population loss. From a technical perspective, this work focuses on *distributed empirical risk minimization with efficient communication complexity*; that is, the objective is to design an efficient distributed protocol which minimizes the empirical loss.

While this deviates from the main objective in statistical learning of minimizing the *population loss*, we focus on the empirical loss for the following reasons:

(i) *Efficient communication implies generalization*: As discussed in (Kane, Livni, Moran, and Yehudayoff, 2019; Braverman, Kol, Moran, and Saxena, 2019), Occam's razor and sample compression arguments can be naturally used to bound the *generalization gap* — i.e. the absolute difference between the empirical and population losses — of efficient distributed learning algorithms. In a nutshell, the bound follows by arguing that the output hypothesis is determined by the communication transcript of the protocol. Hence, the communication complexity of the protocol upper-bounds the *description length* of the output hypothesis, which translates to a bound on the generalization gap via Occam's razor or sample compression. In particular, this reasoning applies to the algorithm we present in this work, and hence it generalizes. Thus, for communication-efficient protocols, the empirical loss is a good proxy of the population loss.

(ii) Focusing on empirical loss simplifies the exposition: while it is possible to translate our results to the setting of population loss, this introduces additional probabilistic machinery and complicates the presentation without introducing any new ideas. Further, Empirical risk minimization is a natural and classical problem, and previous work on distributed PAC learning focused on it, at least implicitly (Kane, Livni, Moran, and Yehudayoff, 2019; Vempala, Wang, and Woodruff, 2020; Braverman, Kol, Moran, and Saxena, 2019).

Paper organization. In Section 2 we formally define the model and give an overview of our results and related work. Section 3 contains brief preliminaries. We prove the upper bound in Section 4 and the lower bound in Section 5. The paper closes with Section 6, which discusses directions for future research.

2. Model and results

2.1. Model

Following (Balcan, Blum, Fine, and Mansour, 2012), we consider a distributed setting consisting of k players numbered $1, \ldots, k$, and a center. Each player can communicate only with the center. An hypothesis class $\mathcal{H}$ over a universe $\mathcal{U}$ is given, and a finite domain set $U \subset \mathcal{U}$ of size n is given as well. We denote the VC-dimension of $\mathcal{H}$ by $d := d(\mathcal{H})$. The finite domain U is known in advance to the center and to all players. A pair $z := (x, y)$, where $x \in U$ and $y \in \{\pm 1\}$, is called an *example*. A sequence of examples $z_1, \ldots, z_m$ is called a *sample*, and denoted by S. For a classifier $f : \mathcal{U} \to \{\pm 1\}$, let $E_S(f)$ denote the number of examples in S that f misclassifies:

$$E_S(f) := \sum_{(x,y)\in S} 1[f(x) \neq y].$$

Let OPT be the number of misclassified examples in S with respect to the best hypothesis in $\mathcal{H}$:

$$OPT = OPT(S, \mathcal{H}) := \min_{h\in\mathcal{H}} E_S(h).$$

In most parts of the paper we require that $OPT \in \mathsf{polylog}\, n$. In our setting, a sample S is *adversarially distributed* between the k players into k subsamples $S_1, \ldots, S_k$. Note that the center gets no input. We use the notation $S = \langle S_i \rangle_{i=1}^k$ to clarify that player i has a fraction S_i of the sample, and concatenating all the S_i's yields the entire input sample S.

The goal is to *learn* $\mathcal{H}$, which we define as follows:

Definition 2.1. Let $\mathcal{H}$ be a concept class over a (possibly infinite) universe $\mathcal{U}$ and let k denote the number of players. For a function $T : \mathbb{N} \to \mathbb{N}$ we say that $\mathcal{H}$ is *learnable under the promise* $OPT \leq T(n)$ if there exists a communication complexity bound $C(d, k, n, m) \in \mathsf{poly}(d, k, \log n, \log m)$ such that for every finite $U \subseteq \mathcal{U}$ of size $n = |U|$, there exists a distributed algorithm $\pi = \pi(U)$ that satisfies the following. For every input sample $S = \langle S_i \rangle_{i=1}^k$ with m examples from U, if $OPT = OPT(S, \mathcal{H}) \leq T(n)$ then

the k parties and the center agree on an output hypothesis f which satisfies $E_S(f) \leq OPT$ with probability at least $\frac{2}{3}$ (over the randomness of the protocol π, when randomized), while transmitting at most $C(d, k, n, m)$ bits.

Let us make a few remarks in order to clarify some choices made in the above definition.

1. **Infinite classes**. The above definition allows one to handle natural infinite classes $\mathcal{H}$ such as Euclidean halfspaces. The finite subdomain $U \subseteq \mathcal{U}$ models a particular instance of the learning task defined by $\mathcal{H}$. For example, if $\mathcal{H}$ is the class of halfspaces in $\mathbb{R}^d$, and we use an encoding of real numbers with B bits, then U consists of all possible $2^{d \cdot B}$ points in $\mathbb{R}^d$ that can be encoded. The universal quantification over U serves to make the definition scalable and independent of the encoding of the input points.

2. **The protocol may depend on $U \subseteq \mathcal{U}$.** This possible dependence reflects the fact that when designing algorithms in practice, one knows how the domain points are being encoded as inputs.[2]

2.2. Results

Our positive result is stated in the following theorem.

Theorem 2.2 (Positive Result)**.** *Let $\mathcal{H}$ be a concept class with $d(\mathcal{H}) < \infty$ and let $T = T(n) \in \mathsf{polylog}\, n$. Then, $\mathcal{H}$ is learnable under the promise $OPT \leq T(n)$, and this is achieved by a simple variant of classical boosting. Furthermore, the algorithm is deterministic and oblivious to T and OPT.*

The protocol we use to prove Theorem 2.2 is a *resilient* version of realizable-case boosting. It is resilient in the sense that it can be applied to any input sample, including samples that are *not* realizable by the class $\mathcal{H}$. Moreover, as long as the input sample is sufficiently close to being realizable, this variant of boosting enjoys similar guarantees as in the fully realizable case. This feature of our protocol is not standard in boosting algorithms in the realizable case, which are typically vulnerable to noise (Dietterich, 2000; Long and Servedio, 2010).

Our protocol can be implemented in the no-center model, in which the players can communicate directly (see (Balcan, Blum, Fine, and Mansour, 2012) for a more thorough discussion of these two models), by having one of the players play the part of the center. It also admits a randomized computationally efficient implementation, assuming an oracle access to a PAC learning algorithm for $\mathcal{H}$ in the centralized setting (see Section 4 for further discussion). On the other hand, the protocol is improper. This is unavoidable: a result by Kane, Livni, Moran, and Yehudayoff (2019) shows that even in the realizable case (i.e. $T(n) = 0$), some VC classes cannot be properly learned by communication-efficient protocols.

As mentioned in the introduction, the positive result of Theorem 2.2 can also be proved by reduction to semi-agnostic learning. However, our direct approach results in a simpler protocol.

The following negative result shows that the assumption $OPT \in \mathsf{polylog}\, n$ made by Theorem 2.2 is necessary for allowing communication-efficient learning, even if the protocol is allowed to be randomized and improper.

Theorem 2.3 (Negative Result)**.** *Let $\mathcal{H} = \{h_n : n \in \mathbb{N}\}$, where $h_n(i) = 1$ if and only if $i = n$, be the class of singletons over $\mathbb{N}$. If $T(n) = \log^{\omega(1)}(n)$ then $\mathcal{H}$ is not learnable under the promise that $OPT \leq T(n)$, even when there are only $k = 2$ players.*

When there are two players, our model is equivalent to the standard two-party communication model (Yao, 1979; Kushilevitz and Nisan, 1996; Rao and Yehudayoff, 2020), in which two players, Alice and Bob, communicate through a direct channel, and this is the setting in which we prove Theorem 2.3.

Our results are in fact more general than stated. The algorithm used to prove Theorem 2.2 outputs a hypothesis making at most OPT many mistakes using $OPT \cdot \mathsf{poly}(d, k, \log m, \log n)$ communication (without having to know OPT in advance). The lower bound used to prove Theorem 2.3 shows that for any value $T(n)$ and for any algorithm that learns the class of singletons there exists an input sample with $OPT \approx T(n)$ on which the communication complexity of the protocol is $\Omega(T(n))$.

2.3. Related Work

Originally, distributed learning was studied from the point of view of parallel computation (a partial list includes (Bshouty, 1997; Collins, Schapire, and Singer, 2002; Zinkevich, Weimer, Smola, and Li, 2010; Long and Servedio, 2011)). The focus was on reducing the time complexity rather than the communication complexity. More recent work aims at minimizing communication (Balcan, Blum, Fine, and Mansour, 2012; Daumé, Phillips, Saha, and Venkatasubramanian, 2012a;b; Blum, Heinecke, and Reyzin, 2021). In (Balcan, Blum, Fine, and Mansour, 2012), privacy aspects of such learning tasks are discussed as well.

A related natural model of distributed learning was proposed in (Balcan, Blum, Fine, and Mansour, 2012). In this model, there are k entities and a center, and each entity i can draw examples from a distribution D_i. The goal is to learn a good hypothesis with respect to the mixture distribution

[2] We remark however that the protocol appearing in Section 4 is uniform in U, that is, it can accept U as an additional input.

$D = \frac{1}{k}\sum_{i=1}^{k} D_i$. The communication topology in this model is a star: all entities can communicate only with the center.

In this work, we consider a slightly different model, studied by Daumé, Phillips, Saha, and Venkatasubramanian (2012b;a); Kane, Livni, Moran, and Yehudayoff (2019); Braverman, Kol, Moran, and Saxena (2019), which we call the *adversarial* model. In this model, a sample S is given and partitioned freely among k players by an adversary. While this model might seem less natural, it is more general than the model of Balcan, Blum, Fine, and Mansour (2012), and our main contribution is a protocol that can be applied to this general model.

The work by Lazarevic and Obradovic (2001) suggested a framework for using boosting in distributed environments. In (Chen, Balcan, and Chau, 2016), a clever analysis of "Smooth Boosting" (Kale, 2007) is used to give an efficient semi-agnostic boosting protocol. Kane, Livni, Moran, and Yehudayoff (2019) characterize which classes can be learned in the distributed and proper setting, and give some bounds for different distributed learning tasks. In (Braverman, Kol, Moran, and Saxena, 2019), tight lower and upper bounds on the communication complexity of learning halfspaces are given, using geometric tools.

3. Preliminaries

We use $\log$ for the base 2 logarithm. Let U be the domain set and let $S \subset U \times \{\pm 1\}$ be a sample. We follow the standard definitions of *empirical loss* and *loss*:

$$L_S(f) := \frac{1}{|S|}\sum_{(x,y)\in S} 1[f(x) \neq y],$$
$$L_p(f) := \Pr_{(x,y)\sim p}[f(x) \neq y],$$

respectively, where p is a probability distribution over $U \times \{\pm 1\}$. We now briefly overview some relevant technical tools.

Boosting. The seminal work of Freund and Schapire (1997) used the AdaBoost algorithm to boost a "weak" learner into a "strong" one. In this work we use a simplified version of AdaBoost (see (Schapire and Freund, 2013)): given a distribution p over a sample S, an *α-weak hypothesis* with respect to p is a hypothesis h which is better than a random guess by an additive factor of α:

$$\Pr_{(x,y)\sim p}[h(x) \neq y] \leq 1/2 - \alpha.$$

The boosting algorithm requires an oracle access to an *α-weak learner*, which is an algorithm that returns α-weak hypotheses. Given such a weak learner, the boosting algorithm operates as follows: it receives as input a sample S, and initializes the weight $W_1(z := (x,y))$ of any example $z \in S$ to be 1. In each iteration $t \in 1,\ldots,T$, it then uses the weak learner to obtain an α-weak hypothesis h_t with respect to the distribution p_t on S, which is defined by the weight function W_t, i.e. the probability of each example z is proportional to $W_t(Z)$. The weights are then updated according to the performance of the weak hypothesis h_t on each example:

$$W_{t+1}(z) = W_t \cdot 2^{-1[h(x)=y]}.$$

After T iterations, the algorithm returns the classifier

$$f = \text{sign}\left(\sum_{t=1}^{T} h_t\right).$$

We have the following upper bound[3] on the value of T required for f to satisfy $E_S(f) = 0$.

Theorem 3.1 (Freund and Schapire (1997))**.** *Let $T \geq 6\log|S|$, and assume that in any iteration t, a hypothesis h_t which is $\left(\frac{1}{2} - \frac{1}{15}\right)$-weak with respect to the current distribution p_t is provided to the variant of AdaBoost described above. Then for any $(x,y) \in S$ we have*

$$\frac{1}{T}\sum_{t=1}^{T} 1[h_t(x) \neq y] \leq 1/3.$$

An immediate corollary is that if f is the classifier returned by AdaBoost and $T \geq 6\log|S|$, then $E_S(f) = 0$.

Small ϵ-approximations. Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{U}}$ be a concept class of VC-dimension $d < \infty$, let p be a distribution over examples in $\mathcal{U} \times \{\pm 1\}$, and let $\epsilon > 0$. The seminal uniform convergence theorem of Vapnik and Chervonenkis (1971) implies that a random i.i.d sample S of size $|S| = O(d/\epsilon^2)$ which is drawn from p satisfies with a positive probability that

$$(\forall h \in \mathcal{H}) : |L_S(h) - L_p(h)| \leq \epsilon.$$

Crucially, note that $|S|$ depends only on d, ϵ. In particular, for every distribution p there exists such a sample in its support.

Communication complexity. Our negative result applies already when there are only two players, in which case our model is equivalent to the standard two-party communication model (Yao, 1979; Kushilevitz and Nisan, 1996). One of the standard problems in the two-party communication model is *set disjointness*. In this problem, Alice gets a string $x \in \{0,1\}^n$, Bob gets a string $y \in \{0,1\}^n$, and the goal is to compute the following function $\text{DISJ}_n(x,y)$:

$$\text{DISJ}_n(x,y) = \begin{cases} 0 & x_i = y_i = 1 \text{ for some i,} \\ 1 & \text{otherwise.} \end{cases}$$

[3] This formulation of the theorem appears explicitly as Lemma 2 in (Kane, Livni, Moran, and Yehudayoff, 2019).

The randomized communication complexity of DISJ_n is known to be large:

Theorem 3.2 (Razborov (1990); Kalyanasundaram and Schintger (1992))**.** *The randomized communication complexity of* DISJ_n *is* $\Theta(n)$.

4. A resilient boosting protocol

In this section we use our boosting variant to prove Theorem 2.2, which follows from the next theorem.

Theorem 4.1. *Let $\mathcal{H}$ be an hypothesis class with VC dimension $d < \infty$, let k be the number of players, and let $T(n) \in \mathsf{polylog}(n)$. The protocol* AccuratelyClassify*, described in Figure 2, is a learning protocol under the promise $OPT \leq T(n)$, and has communication complexity of*

$$O\big(OPT \cdot k \log|S|(d \log n + \log|S|)\big),$$

where S is its input sample. Furthermore, if S contains no contradicting examples (that is, examples $(x,+1),(x,-1)$) then the classifier f which the protocol outputs is consistent (i.e. satisfies $E_S(f) = 0$).

AccuratelyClassify relies on the BoostAttempt protocol, appearing in Figure 1, which is similar to classical boosting.

To prove the theorem, we first argue that if BoostAttempt does not get stuck (i.e. it reaches Item 3 in Figure 1), then it simulates boosting and enjoys the guarantees stated in Theorem 3.1. Then, we take into account what happens when BoostAttempt does get stuck; in this case we adopt the perspective inspired by Impagliazzo's Hardcore lemma to remove a small subsample of the input which is "hard" in the sense that every hypothesis in $\mathcal{H}$ has large error on it. Finally, we analyze the total communication cost of the two protocols.

Lemma 4.2. *If protocol* BoostAttempt*, described in Figure 1, outputs a classifier f, then $E_S(f) = 0$.*

Proof. We show that if BoostAttempt does not stop at step 2(e) of some iteration, then in every iteration t, the provided hypothesis h_t is a $\left(\frac{1}{2} - \frac{1}{50}\right)$-weak hypothesis with respect to the current distribution p_t in the boosting process: recall from the preliminaries that p_t is a distribution on S, which is defined by the weight function W_t, i.e. the probability of each example z is proportional to $W_t(Z)$. To establish the above we use two crucial properties of h_t:

- The hypothesis h_t satisfies

  $$L_{D_t}(h_t) \leq 1/100,$$

  where D_t is the distribution defined in step 2(c), i.e. it is the mixture of the uniform distributions over the S'_i's weighted by $\frac{W_t^{(i)}}{W_t}$.

BoostAttempt: Boosting that may get "stuck"

Setting: There are k players and a center, and $\mathcal{H}$ is a known hypothesis class over a domain U.
Input: A distributed sample $S := \langle S_i \rangle_{i=1}^k$, where $S_i = (x_1^i, y_1^i), \ldots, (x_{|S_i|}^i, y_{|S_i|}^i)$ for $i \in [k]$.
Output: Either all players agree on a classifier $f\colon U \to \{\pm 1\}$ which makes no errors on S, or each player i holds a sample $S'_i \subseteq S_i$ such that the concatenated sample $S' = \langle S'_i \rangle_{i=1}^k$ is not realizable. The center holds S'.

1. **Initialize:** Each player i initializes $W_1(z_j^i) = 1$ for all $1 \leq j \leq |S_i|$.
2. For $t := 1, \ldots, T = \lceil 6 \log|S| \rceil$:
   (a) For all $i \in [k]$, let p_t^i be the distribution over S_i defined by $p_t^i(z_j^i) = \frac{W_t(z_j^i)}{W_t^{(i)}}$, where $W_t^{(i)} = \sum_{1 \leq j \leq |S_i|} W_t(z_j^i)$.
   Each player i sends to the center a $\frac{1}{100}$-approximation w.r.t. p_t^i of minimal size, denoted by $S'_i = \hat{z}_1^i, \ldots, \hat{z}_{|S'_i|}^i$.
   (b) Each player i sends $W_t^{(i)}$ to the center.
   (c) Let $S' = \langle S'_i \rangle_{i=1}^k$. Let D_t be the distribution on S' defined by $D_t(\hat{z}_j^i) = \frac{1}{|S'_i|} \cdot \frac{W_t^{(i)}}{W_t}$, where $W_t = \sum_{i=1}^k W_t^{(i)}$ is the total sum of weights.
   (d) If there is $\hat{h} \in \mathcal{H}$ such that $L_{D_t}(\hat{h}) \leq 1/100$ then:
   - The center sets $h_t := \hat{h}$ and sends h_t to all players.

   (e) Else:
   - Output S'.

   (f) Each player i updates

   $$W_{t+1}(z_j^i) = W_t(z_j^i) \cdot 2^{-1[h_t(x_j^i) = y_j^i]}$$

   for any $z_j^i \in S_i$.
3. Output the classifier

$$f(x) = \mathrm{sign}\left(\sum_{t=1}^{T} h_t(x)\right).$$

Figure 1. A boosting protocol that may get "stuck" when the input sample is not realizable.

AccuratelyClassify: A learning protocol

Setting: There are k players and a center, and $\mathcal{H}$ is a known hypothesis class over a domain U.
Input: A distributed sample $S := \langle S_i \rangle_{i=1}^k$. (Below we treat each S_i as a multiset.)
Output: A classifier $f : U \to \{\pm 1\}$.

1. **Initialize:** The center initializes a multiset $\mathcal{D} := \emptyset$.
2. While BoostAttempt$(\langle S_i \rangle_{i=1}^k)$ returns a non-realizable subsample $S' := \langle S'_i \rangle_{i=1}^k$:
   (a) The center updates $\mathcal{D} := \mathcal{D} \cup S'$.
   (b) Each player updates $S_i := S_i \setminus S'_i$.
3. Let g be the classifier returned by BoostAttempt.
4. For every $x \in U$, let $n_+(x)$ be the number of times that the example $(x, +1)$ occurs in $\mathcal{D}$, and define $n_-(x)$ similarly.
5. Output the classifier $f : U \to \{\pm 1\}$ defined for any $x \in U$ as follows:

$$f(x) = \begin{cases} +1 & n_+(x) \geq 1, n_+(x) \geq n_-(x), \\ -1 & n_-(x) \geq 1, n_-(x) > n_+(x), \\ g(x) & \text{otherwise.} \end{cases}$$

Figure 2. A resilient improper, deterministic learning protocol.

- S'_i is a $\frac{1}{100}$-approximation of the distribution p_t^i on S_i, defined by $p_t^i(z_j^i) = \frac{W_t(z_j^i)}{W_t^{(i)}}$, and hence

$$\left| L_{S'_i}(h) - L_{p_t^i}(h) \right| \leq 1/100$$

for all $h \in \mathcal{H}$.

Let p_t be the normalization of the weights in iteration t, that is $p_t(z_j^i) = \frac{W_t(z_j^i)}{W_t}$. So:

$$\begin{aligned}
L_{p_t}(h_t) &= \sum_{i=1}^k \sum_{z_j^i \in S_i} p_t(z_j^i) 1[h_t(x_j^i) \neq y_j^i] \\
&= \sum_{i=1}^k \sum_{z_j^i \in S_i} \frac{W_t(z_j^i)}{W_t} 1[h_t(x_j^i) \neq y_j^i] \\
&= \sum_{i=1}^k \frac{W_t^{(i)}}{W_t} \sum_{z_j^i \in S_i} \frac{W_t(z_j^i)}{W_t^{(i)}} 1[h_t(x_j^i) \neq y_j^i] \\
&= \sum_{i=1}^k \frac{W_t^{(i)}}{W_t} L_{p_t^i}(h_t) \\
&\leq \sum_{i=1}^k \frac{W_t^{(i)}}{W_t} \left[L_{S'_i}(h_t) + 1/100 \right] \\
&= \sum_{i=1}^k \frac{W_t^{(i)}}{W_t} \left[\frac{\sum_{\hat{z}_j^i \in S'_i} 1[h_t(\hat{x}_j^i) \neq \hat{y}_j^i]}{|S'_i|} + 1/100 \right] \\
&= L_{D_t}(h_t) + 1/100 \\
&\leq 1/100 + 1/100 = 1/50.
\end{aligned}$$

Since $1/50 < 1/15$, by Theorem 3.1 a total of $\lceil 6 \log |S| \rceil$ iterations are enough to output a classifier f that satisfies $E_S(f) = 0$. □

Next, we we consider the case in which BoostAttempt does get stuck. In this case, note that the small sample S' sent to the center is not realizable.

Observation 4.3. Let D be a distribution over a sample S. If for all $h \in \mathcal{H}$ it holds that $L_D(h) > 1/100$ then S is not realizable.

The following observation states that BoostAttempt is called at most OPT times by AccuratelyClassify.

Observation 4.4. Let S be a non-realizable sample, and let S' be a non-realizable subsample of S. Then for all $h \in \mathcal{H}$,

$$E_S(h) > E_{S \setminus S'}(h).$$

That is, if we remove any non-realizable subsample from S, then the *number* of mistakes of any hypothesis decreases by at least 1.

We are now ready to prove Theorem 4.1. The main part is analysing the communication complexity of BoostAttempt.

Theorem 4.1. First we show correctness, and then analyze the communication complexity.

Correctness. The loop in AccuratelyClassify is executed as long as BoostAttempt returns a non-realizable sample. Due to Observation 4.4, after at most OPT iterations, BoostAttempt will return a classifier, since the input sample will then be realizable. This classifier makes zero errors on the input to BoostAttempt, due to Lemma 4.2. Consequently, the classifier f returned by AccuratelyClassify makes the least number of errors among all possible classifiers. Furthermore, if S contains no contradicting examples, then $E_S(f) = 0$.

Communication. We first analyze the communication complexity of BoostAttempt and show that its upper bounded by $O(k \log |S|(d \log n + \log|S|))$. First, it has $\lceil 6 \log|S| \rceil = O(\log|S|)$ iterations. In each iteration, k many $\frac{1}{100}$-approximations are sent to the center in step 2(a), each taking $O(d \log n)$ bits to encode, according to (Vapnik and Chervonenkis, 1971). Then, the sums of weights of each player are sent to the center in step 2(b). This requires $O(k \log|S|)$ communication: indeed, the initial weight of each element is 1, and in each iteration it might be halved. There are $O(\log|S|)$ iterations, so the weight of any element may decrease up to $\Omega(1/|S|)$. So, encoding the sums of weights in step 2(b) requires $O(k \log|S|)$ bits. Steps 2(c-e) can now be executed by the center, with zero communication. Now, if the condition in step 2(d) does not hold, a non-realizable sample S', which is the concatenation of the $\frac{1}{100}$-approximations S'_i, is outputted by BoostAttempt. This step requires k bit of communication, in which the center indicates to each of the players that this condition does not hold. Also notice that this step happens at most once and hence increases the total communication complexity by at most k bits. If this condition holds and the protocol continues, then each player updates its weights with zero communication. Thus, we get a total of $O(k \log |S|(d \log n + \log|S|))$ communication used in BoostAttempt.

AccuratelyClassify executes BoostAttempt at most OPT times due to Observation 4.4, and hence the total communication used by AccuratelyClassify is $O(OPT \cdot k \log|S|(d \log n + \log|S|))$. □

A computationally efficient implementation. We defined BoostAttempt as a communication-efficient deterministic protocol. However, as currently formulated, the protocol is not computationally efficient, since step 2(a) requires finding a $\frac{1}{100}$-approximation, which cannot be done efficiently in general. Vapnik and Chervonenkis (1971) proved that a random sample of size $O(d/\epsilon^2)$ is an ϵ-approximation with high probability. This can be used to make our protocol efficient at the cost of making it randomized. Furthermore, notice that in step 2(d), a weak hypothesis for the distribution D_t on S' is found by the center. This step can also be implemented efficiently provided that $\mathcal{H}$ admits an efficient agnostic PAC learner in the centralized setting.

5. A complementing negative result

In this section we prove Theorem 2.3.

Theorem (Theorem 2.3 restatement). Let $\mathcal{H} = \{h_n : n \in \mathbb{N}\}$, where $h_n(i) = 1$ if and only if $i = n$, be the class of singletons over $\mathbb{N}$. If $T(n) = \log^{\omega(1)} n$ then $\mathcal{H}$ is not learnable under the promise that $OPT \leq T(n)$, even when there are only $k = 2$ players.

The proof uses a mapping suggested in (Kane, Livni, Moran, and Yehudayoff, 2019) together with Theorem 3.2, the well-known communication lower bound for set disjointness.

Lemma 5.1 (Kane, Livni, Moran, and Yehudayoff (2019))**.** *Let $x, y \in \{0,1\}^n$, and let $w(x)$ denote the hamming weight of a binary string x. Let $\mathcal{H}$ be the class of singletons over $[n]$ (it contains exactly all hypotheses that assign 1 to a single $i \in [n]$ and -1 to all other elements). Then, there are mappings $F_a, F_b \colon \{0,1\}^n \to ([n] \times \{\pm 1\})^n$ taking boolean n-vectors to samples such that the combined sample $S := \langle F_a(x); F_b(y) \rangle$ satisfies:*

1. *If $\mathrm{DISJ}_n(x,y) = 1$ then $E_S(f) \geq w(x) + w(y)$ for any classifier f (not necessarily from $\mathcal{H}$).*
2. *If $\mathrm{DISJ}_n(x,y) = 0$ then the optimal $h \in \mathcal{H}$ satisfies $E_S(h) = w(x) + w(y) - 2$.*

The proof follows by letting

$$F_a(x) = \left\{(i, (-1)^{1-x_i}) : i \in [n]\right\},$$
$$F_b(y) = \left\{(i, (-1)^{1-y_i}) : i \in [n]\right\}.$$

Those mappings are used in (Kane, Livni, Moran, and Yehudayoff, 2019) to prove a reduction to set disjointness, in order to show that agnostic classification requires $\Omega(n)$ communication under some conditions. A slight modification of their proof results in the bound of Theorem 2.3.

Proof of Theorem 2.3. Let $n \in \mathbb{N}$ and set $U = [n]$. Given a randomized improper learning protocol $\pi(U)$ for $\mathcal{H}$ under the promise that $OPT \leq T(n)$, we construct the following protocol π' for DISJ_r, where $r = \lfloor \frac{T(n)}{2} \rfloor$.

1. Let $x, y \in \{0,1\}^r$ denote the inputs for DISJ_r.
2. Publish $w(x), w(y)$.

3. Extend x, y to strings $x', y' \in \{0,1\}^n$ by adding $n-r$ zeroes to each.

4. Construct $S := \langle F_a(x'); F_b(y') \rangle$ as described in Lemma 5.1.

5. Execute $\pi(S)$ and let f be the hypothesis it outputs.

6. Output 1 if and only if $E_S(f) \geq w(x) + w(y)$.

Note that by construction, OPT is at most $2r \leq T(n)$ (because any singleton h_i where $i \leq r$ has error at most $2r$ on S). So, $OPT \leq T(n)$ and therefore Lemma 5.1 implies that this protocol solves set disjointness correctly with probability at least $2/3$. Thus, by Theorem 3.2, its communication complexity is $\Omega(r) = \Omega(T(n))$.

We now wrap up the proof by showing that the communication complexity of π is not in

$$\mathsf{poly}(\log n, \log|S| = \log n, k = 2) = \mathsf{polylog}(n).$$

Indeed, the communication complexity of π' is at most $2 \log r$ larger than that of π. Thus, also the communication complexity of π is $\Omega(r) = \Omega(T(n))$, and by assumption $T(n) = \log^{\omega(1)} n$. □

6. Suggestions for future research

Characterizing agnostic learning. Our main result can be viewed as an agnostic learning protocol whose communication complexity depends linearly on OPT. There are concept classes in which such dependence is necessary, as shown by Theorem 2.3. It is also easy to see that there are classes for which this dependence can be avoided, for example finite classes. Is there a natural characterization of those classes which are learnable without any promise on OPT? Are there infinite classes with this property?

The approximation factor in semi-agnostic learning. Balcan, Blum, Fine, and Mansour (2012) and Chen, Balcan, and Chau (2016) give efficient semi-agnostic learners that approximate the error of a best hypothesis from the class up to a multiplicative factor of $c \geq 4$. A simple alteration of the constants in their proofs improves the approximation factor to $2 + \alpha$ for every $\alpha > 0$ (at the cost of higher communication complexity which deteriorates as $\alpha \to 0$). Can the multiplicative factor be further improved, say to c for some $c \leq 2$?

Bounded communication complexity and generalization. It is interesting to further explore the relationship between the communication complexity and the generalization capacity of distributed learning protocols.

Acknowledgments

We thank an anonymous ALT 2022 reviewer for pointing out that Theorem 2.2 can be proved by reduction to semi-agnostic learning.

References

Balcan, M. F., Blum, A., Fine, S., and Mansour, Y. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pp. 26.1–26.22. JMLR Workshop and Conference Proceedings, 2012.

Blum, A., Heinecke, S., and Reyzin, L. Communication-aware collaborative learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

Braverman, M., Kol, G., Moran, S., and Saxena, R. R. Convex set disjointness, distributed learning of halfspaces, and lp feasibility. *arXiv preprint arXiv:1909.03547*, 2019.

Bshouty, N. H. Exact learning of formulas in parallel. *Machine Learning*, 26(1):25–41, 1997.

Bun, M., Kamath, G., Steinke, T., and Wu, S. Z. Private hypothesis selection. *Advances in Neural Information Processing Systems*, 32:156–167, 2019.

Chen, S.-T., Balcan, M.-F., and Chau, D. H. Communication efficient distributed agnostic boosting. In *Artificial Intelligence and Statistics*, pp. 1299–1307. PMLR, 2016.

Collins, M., Schapire, R. E., and Singer, Y. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1):253–285, 2002.

Daumé, III, H., Phillips, J., Saha, A., and Venkatasubramanian, S. Protocols for learning classifiers on distributed data. In *Artificial Intelligence and Statistics*, pp. 282–290. PMLR, 2012a.

Daumé, III, H., Phillips, J. M., Saha, A., and Venkatasubramanian, S. Efficient protocols for distributed classification and optimization. In *International Conference on Algorithmic Learning Theory*, pp. 154–168. Springer, 2012b.

Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Impagliazzo, R. Hard-core distributions for somewhat hard problems. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pp. 538–545. IEEE, 1995.

Kale, S. Boosting and hard-core set constructions: a simplified approach. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, pp. 131. Citeseer, 2007.

Kalyanasundaram, B. and Schintger, G. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

Kane, D., Livni, R., Moran, S., and Yehudayoff, A. On communication complexity of classification problems. In *Conference on Learning Theory*, pp. 1903–1943. PMLR, 2019.

Kushilevitz, E. and Nisan, N. *Communication Complexity*. Cambridge University Press, 1996. doi: 10.1017/CBO9780511574948.

Lazarevic, A. and Obradovic, Z. The distributed boosting algorithm. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 311–316, 2001.

Long, P. and Servedio, R. Algorithms and hardness results for parallel large margin learning. *Advances in Neural Information Processing Systems*, 24:1314–1322, 2011.

Long, P. M. and Servedio, R. A. Random classification noise defeats all convex potential boosters. *Machine learning*, 78(3):287–304, 2010.

Rao, A. and Yehudayoff, A. *Communication Complexity: and Applications*. Cambridge University Press, 2020. doi: 10.1017/9781108671644.

Razborov, A. A. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming*, pp. 249–253. Springer, 1990.

Schapire, R. E. and Freund, Y. Boosting: Foundations and algorithms. *Kybernetes*, 2013.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Vempala, S. S., Wang, R., and Woodruff, D. P. The communication complexity of optimization. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1733–1752. SIAM, 2020.

Yao, A. C.-C. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pp. 209–213, 1979.

Zinkevich, M., Weimer, M., Smola, A. J., and Li, L. Parallelized stochastic gradient descent. In *NIPS*. Citeseer, 2010.