Proceedings of Machine Learning Research vol 134:1–60, 2021 34th Annual Conference on Learning Theory

Boosting in the Presence of Massart Noise

Ilias Diakonikolas ILIAS@CS.WISC.EDU
University of Wisconsin-Madison

Russell Impagliazzo RUSSELL@ENG.UCSD.EDU
Daniel Kane DAKANE@UCSD.EDU
Rex Lei RLEI@ENG.UCSD.EDU
Jessica Sorrell JLSORREL@ENG.UCSD.EDU
University of California-San Diego

Christos Tzamos TZAMOS@WISC.EDU
University of Wisconsin-Madison

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We study the problem of boosting the accuracy of a weak learner in the (distribution-independent) PAC model with Massart noise. In the Massart noise model, the label of each example x is independently misclassified with probability $\eta(x) \leq \eta$, where $\eta < 1/2$. The Massart model lies between the random classification noise model and the agnostic model. Our main positive result is the first computationally efficient boosting algorithm in the presence of Massart noise that achieves misclassification error arbitrarily close to η. Prior to our work, no non-trivial booster was known in this setting. Moreover, we show that this error upper bound is best possible for polynomial-time black-box boosters, under standard cryptographic assumptions. Our upper and lower bounds characterize the complexity of boosting in the distribution-independent PAC model with Massart noise. As a simple application of our positive result, we give the first efficient Massart learner for unions of high-dimensional rectangles.

Keywords: Boosting, Massart Noise, PAC Learning

1. Introduction

1.1. Background and Motivation

Boosting is a general learning technique that combines the outputs of a weak base learner — a learning algorithm with low but non-trivial accuracy — to obtain a hypothesis of higher accuracy. Boosting has been extensively studied in machine learning and statistics since initial work by Schapire (Schapire, 1990). The reader is referred to (Schapire, 2003) for an early survey from the theoretical machine learning community, (Bühlmann and Hothorn, 2007) for a statistics perspective, and (Schapire and Freund, 2012) for a book on the topic. Here we study boosting in the context of learning classes of Boolean functions with a focus on Valiant's distribution-independent PAC model (Valiant, 1984). During the past three decades, several efficient boosting procedures have been developed in the *realizable* PAC model, i.e., when the data is consistent with a function in the target class. On the other hand, boosting in the presence of *noisy data* remains less understood.

In this work, we study the complexity of boosting in the presence of *Massart noise*. In the Massart (or bounded noise) model, the label of each example x is flipped independently with probability $\eta(x) \leq \eta$, for some parameter $\eta < 1/2$. The flipping probability $\eta(x)$ is bounded but is unknown to the learner and can depend on the example x in a potentially adversarial manner. Formally, we have the following definition.

Definition 1 (PAC Learning with Massart Noise) *Let $\mathcal{C}$ be a concept class over $X = \mathbb{R}^n$, D_x be any fixed but unknown distribution over X, and $0 \leq \eta < 1/2$ be the noise parameter. Let $f \in \mathcal{C}$ be the unknown target concept. A* noisy example oracle*, $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, works as follows: Each time $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ is invoked, it returns a labeled example (x, y), where $x \sim D_x$, $y = f(x)$ with probability $1 - \eta(x)$ and $y = -f(x)$ with probability $\eta(x)$, for an* unknown *function $\eta(x) : X \to [0, \eta]$. Let D denote the joint distribution on (x, y) generated by the above oracle. A learning algorithm is given i.i.d. samples from D and its goal is to output a hypothesis h such that with high probability the misclassification error $\mathbf{Pr}_{(x,y)\sim D}[h(x) \neq y]$ is as small as possible. We will use $\mathrm{OPT} \stackrel{\mathrm{def}}{=} \inf_{g \in \mathcal{C}} \mathbf{Pr}_{(x,y)\sim D}[g(x) \neq y]$ to denote the optimal misclassification error.*

Background on Massart Noise. The Massart model is a natural semi-random input model that is more realistic and robust than random classification noise. Noise can reflect computational difficulty or ambiguity, as well as random factors. For example, a cursive "e" might be substantially more likely to be misclassified as "a" than an upper case Roman letter. Massart noise allows for these variations in misclassification rates, while not requiring precise knowledge of which instances are more likely to be misclassified. That is, algorithms that learn in the presence of Massart noise are likely to be less brittle than those that depend on uniformity of misclassification noise. Agnostic learning is of course even more robust, but unfortunately, it can be computationally infeasible to design agnostic learners for many applications.

In its above form, the Massart noise model was defined in (Massart and Nedelec, 2006). An essentially equivalent noise model had been defined in the 80s by Sloan and Rivest (Sloan, 1988, 1992; Rivest and Sloan, 1994; Sloan, 1996), and a very similar definition had been considered even earlier by Vapnik (Vapnik, 1982). The Massart model is a generalization of the Random Classification Noise (RCN) model (Angluin and Laird, 1988) and appears to be easier than the agnostic model (Haussler, 1992; Kearns et al., 1994). Perhaps surprisingly, until very recently, no progress had been made on the efficient, distribution-free PAC learnability in the presence of Massart noise for any non-trivial concept class.

In more detail, the existence of an efficient distribution-independent PAC learning algorithm with non-trivial error guarantee for any concept class in the Massart model had been posed as an open question in a number of works, including (Sloan, 1988; Cohen, 1997), and was highlighted in A. Blum's FOCS'03 tutorial (Blum, 2003). Recent work (Diakonikolas et al., 2019) made the first algorithmic progress in this model for the concept class of halfspaces. Specifically, (Diakonikolas et al., 2019) gave a polynomial-time learning algorithm for Massart halfspaces with misclassification error $\eta + \epsilon$. We note that the information-theoretically optimal error is $\mathrm{OPT} = \mathbb{E}_{x \sim D_x}[\eta(x)]$, which is at most η but could be much smaller. Thus, the error achieved by the aforementioned algorithm can be very far from optimal. Very recent follow-up work (Chen et al., 2020) showed that obtaining the optimal error of $\mathrm{OPT} + \epsilon$ for halfspaces requires super-polynomial time in Kearns' Statistical Query (SQ) model (Kearns, 1998). Contemporaneous to the results of the current paper, (Diakonikolas and Kane, 2020) showed an SQ lower bound ruling out any constant factor or even polynomial factor approximation for this problem. The approximability of learning Massart halfspaces remains a challenging open problem of current investigation.

Comparison to RCN and Agnostic Noise. Random Classification Noise (RCN) (Angluin and Laird, 1988) is the special case of Massart noise where the label of each example is independently flipped with probability *exactly* $\eta < 1/2$. RCN is a fundamentally easier model algorithmically. Roughly speaking, RCN is predictable which allows us to cancel out the effect of the noise on any computation, in expectation. A formalization of this intuition is that *any* Statistical Query (SQ) algorithm (Kearns, 1998) is automatically robust to RCN. This fact inherently fails in the presence of Massart noise. Roughly speaking, the ability of the Massart adversary to choose *whether* to flip a label and if so, with what probability, makes this model algorithmically challenging. Moreover, the uniform noise assumption in the RCN model is commonly accepted to be unrealistic, since in practical scenarios some instances are harder to classify than others (Frénay and Verleysen, 2013). For example, in the setting of human annotation noise (Beigman and Klebanov, 2009), it has been observed that the flipping probabilities are not uniform.

The agnostic model (Haussler, 1992; Kearns et al., 1994) is the most challenging noise model in the literature, in which an adversary can arbitrarily flip an $\mathrm{OPT} < 1/2$ fraction of the labels. It is well-known that (even weak) learning in this model is computationally intractable for simple concept classes, including halfspaces (Daniely, 2016).

The Massart model can be viewed as a reasonable compromise between RCN and the agnostic model, in the sense that it is a realistic noise model that may allow for efficient algorithms in settings where agnostic learning is computationally hard. This holds in particular for the important concept class of halfspaces. As already mentioned, even weak learning of halfspaces is hard in the agnostic model (Daniely, 2016), while an efficient Massart learner with non-trivial accuracy is known (Diakonikolas et al., 2019).

Boosting With Noisy Data. An important research direction, which was asked in Schapire's original paper (Schapire, 1990), is to design boosting algorithms in the presence of noisy data. This broad question has been studied in the past two decades by several researchers. See Section 1.4 for a detailed summary of related work. Specifically, prior work has obtained efficient boosters for RCN (Kalai and Servedio, 2003) and agnostic noise (Servedio, 2003; Feldman, 2010). It should be emphasized that these prior works do not immediately extend to give boosters for the Massart noise setting. For example, while the agnostic model is stronger than the Massart model, an agnostic booster does not imply a Massart booster, as it relies on a much stronger assumption — the existence of a weak *agnostic* learner. That is, the complexity of noisy boosting is not "monotone" in the difficulty of the underlying noise model. More broadly, it turns out that the complexity of boosting with inconsistent data, and the underlying boosting algorithms, crucially depend on the choice of the noise model.

In this work, we ask the following question:

Can we develop efficient boosting algorithms for PAC learning with Massart noise?

Our focus is on the distribution-independent setting. Given a distribution-independent Massart weak learner for a concept class $\mathcal{C}$, we want to design a distribution-independent Massart learner for $\mathcal{C}$ with high(er) accuracy. Prior to this work, no progress had been made on this front. *In this paper, we resolve the complexity of the aforementioned problem by providing (1) an efficient boosting algorithm and (2) a matching computational lower bound on the error rate of any black-box booster.*

This work is the first step of the broader agenda of developing a general algorithmic theory of boosting for other "benign" semi-random noise models, lying between random and fully adversarial corruptions.

1.2. Our Results

Our main result is the first computationally efficient boosting algorithm for distribution-independent PAC learning in the presence of Massart noise that guarantees misclassification arbitrarily close to η, where η is the upper bound on the Massart noise rate. To state our main result, we will require the definition of a Massart weak learner (see Definition 10 for additional details).

Definition 2 (Massart Weak Learner) *Let $\alpha, \gamma \in (0, 1/2)$. An (α, γ)-Massart weak learner* `WkL` *for concept class $\mathcal{C}$ is an algorithm that, for any distribution D_x over examples, any function $f \in \mathcal{C}$, and any noise function $\eta(x)$ with noise bound $\eta < 1/2 - \alpha$, outputs a hypothesis h that with high probability satisfies $\mathbf{Pr}_{(x,y)\sim D}[h(x) \neq y] \leq 1/2 - \gamma$, where D is the joint Massart noise distribution.*

We prove two versions of our main algorithmic result. In Section 3, we present our Massart noise-tolerant booster (Algorithm 1). In Appendix B, we analyze this algorithm and show that it converges within $O(1/(\eta\gamma^2))$ rounds of boosting (Theorem B.1). In Section C, we give a more careful analysis of convergence, showing that the same algorithm in fact converges in $O(\log^2(1/\eta)/\gamma^2)$ rounds (Theorem C.1). In fact, the latter upper bound is nearly optimal for distribution-independent boosters (see, e.g., Chapter 13 of (Schapire and Freund, 2012)). We now state our main result:

Theorem 3 (Main Result) *There exists an algorithm* `Massart-Boost` *that for every concept class $\mathcal{C}$, given samples to a Massart noise oracle $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, where $f \in \mathcal{C}$, and black-box access to an (α, γ)-Massart weak learner* `WkL` *for $\mathcal{C}$,* `Massart-Boost` *efficiently computes a hypothesis h that with high probability satisfies $\mathbf{Pr}_{(x,y)\sim D}[h(x) \neq y] \leq \eta(1 + O(\alpha))$. Specifically,* `Massart-Boost` *makes $O(\log^2(1/\eta)/\gamma^2)$ calls to* `WkL` *and draws*

$$\mathrm{polylog}(1/(\eta\gamma))/(\eta\gamma^2)\ m_{\mathtt{WkL}} + \mathrm{poly}(1/\alpha, 1/\gamma, 1/\eta)$$

samples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, where $m_{\mathtt{WkL}}$ is the number of samples required by `WkL`.

Prior to this work, no such boosting algorithm was known for PAC learning with Massart noise. Moreover, as we explain in Section 1.4, previous noise-tolerant boosters do not extend to the Massart noise setting. In Section 1.3, we provide a detailed overview of our new algorithmic ideas to achieve this.

Some additional comments are in order. First, we note that the $\eta + \epsilon$ error guarantee achieved by our efficient booster can be far from the information-theoretic minimum of $\mathrm{OPT} + \epsilon$. The error guarantee of our generic booster matches the error guarantee of the best known polynomial-time learning algorithm for Massart halfspaces (Diakonikolas et al., 2019). Interestingly, the learning algorithm of (Diakonikolas et al., 2019) can be viewed as a specialized boosting algorithm for the class of halfspaces, in which the halfspace structure is used to downweight specific regions on which the current classifier achieves high accuracy. Theorem 3 is a broad generalization of this result that applies to *any* concept class. This connection was one of the initial motivations for this work.

A natural question is whether the error upper bound achieved by our booster can be improved. Perhaps surprisingly, we show that our guarantee is best possible for black-box boosting algorithms (under cryptographic assumptions). Specifically, we have the following theorem:

Theorem 4 (Lower Bound on Error of Black-Box Massart Boosting) *Assuming one-way functions exist, no polynomial-time boosting algorithm, given black-box access to an (α, γ)-Massart weak learner, can output a hypothesis h with misclassification error $\mathbf{Pr}_{(x,y)\sim D}[h(x) \neq y] \leq \eta(1 + o(\alpha))$, where η is the upper bound on the Massart noise rate. In particular, this statement remains true on Massart distributions with optimal misclassification error $\mathrm{OPT} \ll \eta$.*

The reader is referred to Theorem 17 for a detailed formal statement. Our lower bound establishes that the error upper bound achieved by our boosting algorithm is best possible. It is worth pointing out a related lower bound shown in (Kalai and Servedio, 2003) in the context of RCN. Specifically, (Kalai and Servedio, 2003) showed that any efficient black-box booster tolerant to RCN must incur error at least η (with respect to the target function f), where η is the RCN noise rate. Since RCN is the special case of Massart noise where $\eta(x) = \eta$ for all x, the lower bound of (Kalai and Servedio, 2003) suggests a lower bound of OPT for black-box Massart boosting. Importantly, our lower bound is significantly stronger, as it shows a lower bound of η, even when OPT is much smaller than η.

Intriguingly, Theorem 4 shows that the error guarantee of the (Diakonikolas et al., 2019) learning algorithm for Massart halfspaces cannot be improved using boosting, and ties with recent work (Diakonikolas and Kane, 2020) providing evidence that learning with Massart noise (within error relative to OPT) is computationally hard.

Application: Massart Learning of Unions of Rectangles. As an application of Theorem 3, we give the first efficient learning algorithm for unions of (axis-aligned) rectangles in the presence of Massart noise. Interestingly, weak agnostic learning of a single rectangle is computationally hard in the agnostic model (see, e.g., (Feldman et al., 2009)). Recall that a rectangle $R \in \mathbb{R}^d$ is an intersection of inequalities of the form $x \cdot v < t$, where $v \in \{\pm e_j : j \in [d]\}$ and $t \in \mathbb{R}$. Formally, we show:

Theorem 5 *There exists an efficient algorithm that learns unions of k rectangles on $\mathbb{R}^d$ with Massart noise bounded by η. The algorithm has sample complexity $kd^{O(k)}\mathrm{poly}(1/\epsilon, 1/\eta)$, runs in time $(kd^k/\epsilon)^{O(k)}\mathrm{poly}(1/\eta)$, and achieves misclassification error $\eta + \epsilon$, for any $\epsilon > 0$.*

See Theorem 57 for a more detailed statement. Theorem 5 follows by an application of Theorem 3 coupled with a simple weak learner for unions of rectangles that we develop. Our weak learner finds a rectangle entirely contained in the negative region to gain some advantage over a random guess.

It is worth pointing out that the Massart SQ lower bound of (Chen et al., 2020) applies to learning monotone conjunctions. This rules out efficient SQ algorithms with error $\mathrm{OPT} + \epsilon$, even for a single rectangle.

1.3. Overview of Techniques

In this section, we provide a brief overview of our approach.

Boosting Algorithm Approach. We start with our Massart boosting algorithm. Let D be the Massart distribution $\mathtt{Mas}\{f, D_x, \eta(x)\}$ from which our examples are drawn. The distribution D_x on examples is fixed but arbitrary and the function $\eta(x)$ is a Massart noise function satisfying $\eta(x) \leq \eta < 1/2$ with respect to the target function $f \in \mathcal{C}$. As is standard in distribution-independent boosting, our boosting algorithm adaptively generates a sequence of distributions $D^{(i)}$, invokes the weak learner on samples from these distributions, and incrementally combines the corresponding weak hypotheses to obtain a hypothesis with higher accuracy.

The technical challenge of distribution-independent boosting is the adaptive generation of new distributions $D^{(i)}$ that effectively use the weak learner to acquire new and useful information about the target function f. To see why this requires some care, consider an adversarial weak learner that attempts to give the booster as little information about f as possible, while still satisfying its definition as a weak learner. Such an adversarial weak learner might, whenever possible, produce hypotheses that correctly classify the same, small set of examples P, while classifying all other examples randomly. Assuming the function f is balanced, and the intermediate distributions $D^{(i)}$ assign probability at least γ to P, these adversarial hypotheses will have accuracy $1/2 + \gamma$ on their corresponding distributions, while providing no new information about the target function to the booster. To thwart this behavior, the booster must eventually restrict its distributions to assign sufficiently small probability to P to ensure that the weak learner can no longer meet its promised accuracy lower-bound by correctly classifying only the set P. In this way, the booster can force the weak learner to output hypotheses correlating with f on other subsets of its domain. Under reasonable conditions on the specific strategy for reweighting distributions, boosters that incrementally decrease the probability assigned to examples as they are more frequently correctly classified by weak hypotheses are known to eventually converge to high-accuracy hypotheses, by reduction to iterated two-player zero-sum games (Freund and Schapire, 1997b). This general approach to reweighting intermediate distributions is common to all distribution-independent boosters, even in the noiseless setting.

Our booster follows the smooth boosting framework (Servedio, 2003) with some crucial modifications that are necessary to handle Massart noise. A smooth boosting algorithm generates intermediate distributions that do not put too much weight on any individual point, and so do not compel the weak learner to generate hypotheses having good correlation only with noisy examples. This makes the smooth boosting framework a natural starting point for the design of a Massart noise-tolerant booster, though smoothness of the intermediate distributions alone is not a sufficient condition for preservation of the Massart noise property.

To see why, note that to preserve the Massart noise property of the intermediate distributions, it is not enough to enforce an upper bound on the probability that any (potentially noisy) example can be assigned. We require an upper bound on the *relative* probabilities of sampling noisy and correct labels for a given point, to ensure we always have a noise upper bound $\eta^{(i)} < 1/2$. This seems to suggest that preserving the Massart noise property requires a corresponding lower bound on the probability assigned to any given example, so that we do not inadvertently assign more probability to $(x, -f(x))$ than $(x, f(x))$. This is at odds with our strategy for making use of an adversarial weak learner, since guaranteeing progress requires that our distributions can assign arbitrarily small probability to some examples. So, we must use alternative techniques to manage noise.[1]

The fix for this is to simply not include examples (x, y) in the support of $D^{(i)}$ whenever including them could violate the Massart noise property or permit an adversarial weak learner to tell us only what we already know. If many of the weak hypotheses obtained by our booster agree with the label y on x, then we learn little from a marginal weak hypothesis that agrees with y on x. So, we exclude (x, y) from the support of $D^{(i)}$. We must also symmetrically exclude $(x, -y)$, otherwise we

1. We note that vanilla smooth boosting has been shown to succeed in the agnostic model. Interestingly, the above subtle issue for Massart boosting does not arise in agnostic boosting, since agnostic noise is easy to preserve.

risk violating the Massart noise property for $D^{(i)}$, since we have assigned no probability to (x, y), and it may be the case that $-y \neq f(x)$. Withholding these examples allows the booster to get new information from the weak learner in each round, without ever invoking it on an excessively noisy sample.

This balance comes at the cost of updates from the weak learner on withheld examples. This may not seem to pose a significant problem for our booster at first. After all, points on which many hypotheses agree are points where our algorithm is already fairly confident about the correct value of $f(x)$. Unfortunately, this confidence may not be sufficiently justified to ensure an $\eta + \epsilon$ error at the end of the day. In order to deal with this, our algorithm will need to make use of one further idea. We directly check the empirical error of our aggregated hypotheses on the set of withheld examples. If this error is too large (i.e., larger than $\eta+\epsilon$), we conclude we are "overconfident" and have more to learn about the withheld examples after all. Since even an adversarial weak learner will give us new information about these examples in expectation, we include them in subsequent distributions, with appropriate upper and lower bounds on their probabilities to preserve the Massart noise property. If the empirical error is not too large, we are content to learn nothing new about these examples, and so continue to withhold them for the next round of boosting.

Overall, our algorithm will alternate between the two steps of applying the weak learner to an appropriately reweighted version of the underlying distribution, and checking the consistency of our hypotheses with the set of withheld examples. Each step will allow us to make progress in the sense of decreasing a relevant potential function. We iterate these steps until almost all points are consistently being withheld from the weak learner. Once we reach this condition, we will have produced a hypothesis with appropriately small error, and can terminate the algorithm. We analyze the convergence of our algorithm to a low-error hypothesis via a novel potential function that can be easily adapted to analyze other smooth boosting algorithms.

Error Lower Bound. We show that no "black-box" generic boosting algorithm for Massart noise can have significantly better error than that for our algorithm, i.e., $\eta + \Theta(\eta\alpha)$. While this seemingly matches the lower bound for RCN boosting from Kalai and Servedio (2003), the RCN bound only implies a lower bound for RCN weak learners in the special case of Massart noise when $\eta = \mathrm{OPT}$. We show a similar lower bound in the Massart noise setting for a small but polynomial value of OPT. That is, Massart noise boosting algorithms cannot be improved even when only a very small fraction of instances are actually noisy.

To prove our lower bound, we consider a situation where the function to be learned is highly biased, and there is a tiny fraction of inputs with the majority value that are noisy and indistinguishable from non-noisy inputs. If the distribution queried by a boosting algorithm does not reweight values in some way to favor the minority answer, an uncooperative weak learner can return the majority answer and have advantage γ. On the other hand, if the boosting algorithm does reweight values, it risks adding too much noise to the small fraction of already noisy examples, violating the Massart condition. Specifically, we exhibit an adversarial weak learner `rWkL` that has a stability property we call *reproducibility*. `rWkL` returns a hypothesis h that outputs the maximum likelihood label for each heavy-hitter of given distribution D' and outputs a constant value for non-heavy-hitters. Using reproducibility, we argue that i) boosting with `rWkL` can be efficiently simulated without knowing the function f and ii) `rWkL` satisfies the definition of a Massart noise weak learner. We conclude that a black-box boosting algorithm must be able to efficiently learn pseudorandom functions in order to extract useful information from `rWkL`.

1.4. Comparison with Prior Work

The literature on boosting is fairly extensive. Since the initial work of Schapire (Schapire, 1990), boosting has become one of the most studied areas in machine learning — encompassing both theory and practice. Early boosting algorithms (Schapire, 1990; Freund, 1995; Freund and Schapire, 1997a) were not tolerant in the presence of noisy data. In this section, we summarize the most relevant prior work with a focus on boosting techniques that have provable noise tolerance guarantees.

Efficient boosting algorithms have been developed for PAC learning in the agnostic model (Haussler, 1992; Kearns et al., 1994) and in the presence of Random Classification Noise (RCN) (Angluin and Laird, 1988). The notion of agnostic boosting was introduced in (Ben-David et al., 2001). Subsequently, a line of work (Servedio, 2003; Gavinsky, 2003; Kalai et al., 2008; Kalai and Kanade, 2009; Feldman, 2010) developed efficient agnostic boosters with improved error guarantees, culminating in the optimal bound. These agnostic boosters rely on one of two techniques: smooth boosting, introduced in (Servedio, 2003), or boosting via branching programs, developed in (Mansour and McAllester, 2002). While both of these techniques have been successful in the agnostic model, known RCN-tolerant boosters from (Kalai and Servedio, 2003; Long and Servedio, 2005, 2008) are all based on the branching program technique (Mansour and McAllester, 2002). In the following paragraphs, we briefly summarize these two techniques.

Smooth boosting (Servedio, 2003) is a technique that produces intermediate distributions which do not assign too much weight on any single example. The technique was inspired by Impagliazzo's hard-core set constructions in complexity theory (Impagliazzo, 1995) (see also (Klivans and Servedio, 1999; Holenstein, 2005; Barak et al., 2009)) and is closely related to convex optimization. Roughly speaking, smooth boosting algorithms are reminiscent of first-order methods in convex optimization. Smooth boosting methods have been shown to be tolerant to agnostic noise (Servedio, 2003; Gavinsky, 2003; Kalai and Kanade, 2009; Feldman, 2010). Interestingly, (Long and Servedio, 2010) established a lower bound against potential-based convex boosting techniques in the presence of RCN. While we do not prove any relevant theorems here, we believe that our technique can be adapted to give an efficient booster in the presence of RCN.

Another important boosting technique relies on branching programs (Mansour and McAllester, 2002). The main idea is to iteratively construct a branching program in which each internal node is labeled with a hypothesis generated by some call to the weak learner. This technique is quite general and has led to noise tolerant boosters for both RCN (Kalai and Servedio, 2003) (see also (Long and Servedio, 2005, 2008) for refined and simplified boosters relying on this technique) and agnostic noise (Kalai et al., 2008). Roughly speaking, the branching programs methodology leads to "non-convex algorithms" and is quite flexible.

It is worth pointing out that the aforementioned branching program-based boosters do not succeed with Massart noise in their current form. Specifically, the RCN booster in (Kalai and Servedio, 2003) crucially relies on the uniform noise property of RCN, which implies that agreement with the true target function is proportional to agreement with the observed labels. On the other hand, for the agnostic booster of (Kalai et al., 2008), the generated distributions on which the weak learner is invoked do not preserve the Massart noise property — a crucial requirement for any such booster. While it should be possible to adapt the branching program technique to work in the Massart noise model, we believe that the smooth-boosting technique developed in this paper leads to simpler and significantly more efficient boosters that are potentially practical.

Finally, we acknowledge existing work developing efficient learning algorithms for Massart halfspaces (and related noise models) in the *distribution-specific* PAC model (Awasthi et al., 2015, 2016; Zhang et al., 2017; Diakonikolas et al., 2020b; Zhang et al., 2020; Diakonikolas et al., 2020c,a). These works are technically orthogonal to the results of this paper, as they crucially leverage a priori structural information about the distribution on examples (e.g., log-concavity).

1.5. Organization

The structure of this paper is as follows: Section 2 contains preliminary definitions and fixes notation. In Section 3 and Appendix B we present our Massart noise-tolerant boosting algorithm. In Appendix C, we prove an improved round complexity for our booster. In Section 4 and Appendix D, we show that the error achieved by our booster is optimal by proving a lower-bound on the error of any black-box Massart-noise-tolerant booster. In Appendix E, we give an application of our boosting algorithm to learning unions of rectangles. In Appendix F, we give a glossary of symbols.

2. Preliminaries

Throughout this work, we are primarily concerned with large finite domains $\mathcal{X}$. For a distribution D over $\mathcal{X}$, let supp(D) be the set of all $x \in \mathcal{X}$ such that $D(x) \neq 0$. Let $S \mid\mid z$ denote appending z to sequence S. For $f : \mathcal{X} \to \mathbb{R}$, $x \in \mathcal{X}$, we define $\mathrm{sign}(f)(x) = 1$ if $f(x) \geq 0$ and -1 otherwise.

2.1. Massart Noise Model

Let $\mathcal{C}$ be a class of Boolean-valued functions over some domain $\mathcal{X}$, and let D_x be a distribution over $\mathcal{X}$. Let $f \in \mathcal{C}$ be an unknown target function, and let $\eta(x) : \mathcal{X} \to [0, 1/2)$ be an unknown function.

Definition 6 (Noisy Example Oracle) *When invoked, noisy example oracle* $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ *produces a labeled example* (x, y) *as follows:* $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ *draws* $x \sim D_x$*. With probability* $\eta(x)$*, it returns* $(x, -f(x))$*, and otherwise returns* $(x, f(x))$*.*

Definition 7 (Massart Distribution) *A Massart distribution* $D = \mathtt{Mas}\{f, D_x, \eta(x)\}$ *over* $(\mathcal{X}, \pm 1)$ *is the distribution induced by sampling from* $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$*.*

We refer to $\eta(x)$ in this context as the *Massart noise function*. We say a Massart distribution D has *noise rate* η if $\eta(x) \leq \eta$ for all $x \in \mathrm{supp}(D_x)$. The *noise bound* of a Massart noise function is η if $\max_{x \in \mathrm{supp}(D_x)} \eta(x) = \eta$. We emphasize that this model restricts the noise bound to be $\eta < 1/2$.

2.2. Learning under Massart Noise

Let $f : \mathcal{X} \to \{\pm 1\}$ be a function in concept class $\mathcal{C}$. Let $D = \mathtt{Mas}\{f, D_x, \eta(x)\}$ be a Massart distribution over $\mathcal{X}$.

Definition 8 (Misclassification Error, Function Error) *The misclassification error of hypothesis* $h : \mathcal{X} \to \{\pm 1\}$ *over* D *is* $\mathrm{err}_{0\text{-}1}^{D}(h) = \mathbf{Pr}_{(x,y)\sim D}[h(x) \neq y]$*. The error of hypothesis* $h : \mathcal{X} \to \{\pm 1\}$ *with respect to* f *over* D *is* $\mathrm{err}_{0\text{-}1}^{D_x, f}(h) = \mathbf{Pr}_{x \sim D_x}[h(x) \neq f(x)]$*.*

Definition 9 (Advantage) *Hypothesis* $h : \mathcal{X} \to \{\pm 1\}$ *has advantage* $\gamma > 0$ *against distribution* D *if* $\mathrm{err}_{0\text{-}1}^{D}(h) \leq 1/2 - \gamma$*.*

We use the notation $\mathrm{adv}^{D}(h)$ to denote the largest $\gamma \in [0, 1/2]$ for which $\mathrm{err}_{0\text{-}1}^{D}(h) \leq 1/2 - \gamma$.

2.3. Boosting and Weak Learners

Definition 10 (Massart Noise Weak Learner) *Let $\mathcal{C}$ be a concept class of functions $f : \mathcal{X} \to \{\pm 1\}$. Let $\alpha \in [0, 1/2)$. Let $\gamma : \mathbb{R} \to \mathbb{R}$ be a function of α. A Massart noise (α, γ)-weak learner* `WkL` *for $\mathcal{C}$ is an algorithm such that, for any distribution D_x over $\mathcal{X}$, function $f \in \mathcal{C}$, and noise function $\eta(x)$ with noise bound $\eta < 1/2 - \alpha$,* `WkL` *outputs a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ such that $\mathbf{Pr}_S[\mathrm{adv}^D(h) \geq \gamma] \geq 2/3$, where the sample S is drawn from Massart distribution $D = \mathtt{Mas}\{f, D_x, \eta(x)\}$.*

We let γ be a function of α because a given weak learning algorithm may satisfy stronger advantage guarantees if its input distributions are less noisy. For example, the advantage guarantee for our rectangle weak learner (Appendix E) depends quadratically on α.

Our boosting algorithm operates in the *filtering* model of (Bradley and Schapire, 2007). It generates intermediate distributions by drawing examples (x, y) from its oracle $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ and keeping them with probability $\mu(x, y)$, according to a function $\mu : \mathcal{X} \times \{\pm 1\} \to [0, 1]$. We refer to μ informally as a *measure* to emphasize that it induces a distribution D_μ (Definition 22), but need not be one itself. The expectation of the μ with respect to D is a useful quantity for analyzing distribution-independent boosting algorithms. It affects the sample complexity of making calls to the weak learner and is used to bound the error of the final hypothesis output by our algorithm.

Definition 11 (Density of a measure) *Let D be a Massart distribution, and let μ be a measure. The* density *of μ with respect to D is $d(\mu) = \mathbb{E}_{(x,y)\sim D}[\mu(x, y)]$.*

3. Boosting Algorithm

In this section, we describe our Massart noise-tolerant boosting algorithm `Massart-Boost` (Algorithm 1) and state our boosting theorem. More detailed pseudocode can be found in Appendix B, along with a proof of convergence and sample complexity bounds. A proof of the tighter round and sample complexity bounds in Theorem 16 below can be found in Appendix C.

Our algorithm maintains a working hypothesis $\mathrm{sign}(G)$ for $G : \mathcal{X} \to \mathbb{R}$, initialized to 0. We use G to determine measure $\mu(x, y)$, which induces a distribution D_μ on which we query the weak learner. We update G with the resulting weak hypothesis, defining a new distribution over examples by decreasing μ on examples (x, y) for which $\mathrm{sign}(G(x)) = y$ and increasing μ otherwise.

However, to preserve the Massart noise property, we must guarantee μ never assigns too little weight to an example $(x, f(x))$ relative to the weight of $(x, -f(x))$. To ensure we maintain a noise bound below $1/2 - \alpha$ for all intermediate distributions D_μ, we define $\mu(x, y) = 0$ for any example (x, y) at risk of violating this constraint. Since we have defined $\mu(x, y)$ to be anticorrelated with $yG(x)$, noisy example $(x, -f(x))$ may dominate clean example $(x, f(x))$ when $|G(x)|$ is large and $\mathrm{sign}(G(x)) = f(x)$. To preserve the Massart noise property, we pick a threshold s for $|G(x)|$ and partition $\mathcal{X}$ into two sets, $\mathcal{X}^r$ and $\mathcal{X}^s$, based on whether $|G(x)| \geq s$.

The set $\mathcal{X}^r$ contains all $x \in \mathcal{X}$ for which $|G(x)| \geq s$. These x may have high effective noise rate, so they "risk" violating the Massart noise property. Thus, we assign $\mu(x, y) = 0$ for all $x \in \mathcal{X}^r$, regardless of y, removing them from the support of D_μ. The set $\mathcal{X}^s$ contains all $x \in \mathcal{X}$ for which $|G(x)| < s$, ensuring that the effective noise rate of each $x \in \mathcal{X}^s$ (in distribution D_μ) is bounded above by $1/2 - \alpha$. Thus, it is "safe" to call the weak learner on examples (x, y) where $x \in \mathcal{X}^s$. Initially, all $x \in \mathcal{X}$ are in $\mathcal{X}^s$. Our weak hypothesis h is guaranteed to have advantage against D_μ, so we can use h to improve our predictions $G(x)$ for $x \in \mathcal{X}^s$.

To improve our predictions on $x \in \mathcal{X}^r$, `Massart-Boost` performs an additional calibration step. If the working hypothesis $\mathrm{sign}(G)$ misclassifies too many risky examples, it must be "overconfident" in its predictions on $\mathcal{X}^r$; so, we can improve G by adding hypothesis $-\mathrm{sign}(G)$ (similar to the balancing step of (Feldman, 2010)). This recalibration step decreases $|G(x)|$ for $x \in \mathcal{X}^r$, moves all $x \in \mathcal{X}^r$ back into $\mathcal{X}^s$, and allows us to again call the weak learner on these examples. As more examples are correctly classified by $\mathrm{sign}(G)$, the density of the measure μ decreases. When this density is small, the algorithm terminates and returns the classifier $\mathrm{sign}(G)$.

3.1. Definitions

Let function $M : \mathbb{R} \to [0,1]$ satisfy $M(v) = 1$ when $v < 0$, and $M(v) = e^{-v}$ when $v \geq 0$. This "base" measure function is used to both define μ, the measure function used for reweighting intermediate distributions D_μ, and Φ, the potential function used to analyze convergence. We consider a measure function $\mu : \mathcal{X} \times \{\pm 1\} \to [0,1]$ that is parameterized by $s \in \mathbb{R}_{>0}$ and a real-valued function $F : \mathcal{X} \to \mathbb{R}$. In particular, μ assigns no weight to examples (x, y) such that $|F(x)| \geq s$.

Definition 12 (Measure function) *Let* $\mu_{F,s}(x,y) \overset{\text{def}}{=} M(yF(x))$ *if* $|F(x)| < s$ *and* 0 *otherwise.*

At the start of the algorithm, we set $s = \log\left(\frac{1-\eta}{\eta+c}\right)$, where $c = \frac{4\eta\alpha}{1-2\alpha}$. This ensures that the noise rate of distribution D_μ (see Definition 22) is at most $1/2 - \alpha$. Our algorithm uses the measure function $\mu_{G_t,s}$ in round $t \in [T]$, where G_t is the working hypothesis G after t rounds of boosting. To simplify notation, let $\mu_t(x,y) \overset{\text{def}}{=} M(yG_t(x))$ if $|G_t(x)| < s$ and 0 otherwise.

In each round of boosting, we use G_t and s to partition the domain $\mathcal{X}$ into two sets: $\mathcal{X}_t^s$ and $\mathcal{X}_t^r$. If it is "safe" to run the weak learner on a sample containing x, we say $x \in \mathcal{X}_t^s$. Otherwise, $x \in \mathcal{X}_t^r$.

Definition 13 ($\mathcal{X}_t^s$, $\mathcal{X}_t^r$) *For all* $x \in \mathcal{X}$, $x \in \mathcal{X}_t^s$ *if* $|G_t(x)| < s$ *and* $x \in \mathcal{X}_t^r$ *if* $|G_t(x)| \geq s$.

3.2. Pseudocode and Boosting Theorem

Algorithm 1 `Massart-Boost`$^{\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x)), \texttt{WkL}}(\eta, \epsilon, \gamma)$

η: Massart noise rate, ϵ: Target error in excess of η, γ: Weak learner advantage guarantee

$G \leftarrow 0$,
while $d(\mu) > \eta$ **do**
  $S \leftarrow$ sample from D_μ
  $h \leftarrow \texttt{WkL}(S)$
  $h^s(x) \leftarrow h(x)$ if $x \in \mathcal{X}_G^s$ and 0 otherwise
  $G \leftarrow G + \lambda h^s$
  if error of $\mathrm{sign}(G)$ on $\mathcal{X}_G^r$ exceeds $\eta + \epsilon$ **then**
    $h^r(x) \leftarrow -\mathrm{sign}(G(x))$ if $x \in \mathcal{X}_G^r$ and 0 otherwise
    $G \leftarrow G + \lambda h^r$
  update μ according to Definition 12
$H \leftarrow \mathrm{sign}(G_t)$
return H

Algorithm 1 is a simplified psuedocode description of our Massart noise-tolerant boosting algorithm. To prove that Algorithm 1 converges, we show that it make progress in each round of boosting against the following potential function (Lemma 14).

$$\Phi(t) = \mathop{\mathbb{E}}_{(x,y)\sim D} \int_{yG_t(x)}^{\infty} M(z)dz.$$

To see how this function allows us to capture the incremental progress made at each round, consider how the potential $\Phi(t)$ changes as we take a step of size λ in hypothesis space in the direction of some hypothesis h, starting from G_t. If we take λ sufficiently small, then we have from the mean value theorem that the change in potential should be not too much smaller than $\mathbb{E}_{(x,y)\sim D} \lambda y h_t(x) M(yG_t(x))$. Hypothetically, if the function $\mu_{G,s}(x,y)$ used for reweighting were *exactly* $M(yG(x))$, then a hypothesis h with advantage γ would guarantee a drop in potential of roughly

$$\mathop{\mathbb{E}}_{(x,y)\sim D} \lambda y h_t(x)\mu_t(x,y) = \mathop{\mathbb{E}}_{(x,y)\sim D} \lambda y h_t(x) D_{\mu_t}(x,y) d(\mu_t) = \lambda\gamma d(\mu_t).$$

In other words, the advantage guarantee of the weak learner ensures that the potential will drop by an amount proportional to the density of the current measure μ_t. Because $d(\mu_0) = 1$, and the algorithm terminates once $d(\mu_t)$ falls below η, this guaranteed potential drop would allow us to prove termination.

We cannot take $\mu_t(x,y) = M(yG_t(x))$ for all (x,y), however, because such a measure function might overweight noisy examples. Instead, our measure function is exactly $M(yG_t(x))$ only on examples in $\mathcal{X}_t^s$. The proof idea above then guarantees progress on such examples. If an example is not in $\mathcal{X}_t^s$, we permit the algorithm to make no progress or even regress. But, progress is made in expectation over all examples in $\mathcal{X}_t^r$, implying the following lemma (proved in Appendix B).

Lemma 14 (Potential Drop) *Take* $\lambda = \gamma/8$, $\delta_{\texttt{WkL}} = \delta\eta\gamma^2/1536$, *and assume* $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$. *Then for every round of boosting t, with all but probability* $\delta\eta\gamma^2/768$,

$$\Phi(t) - \Phi(t+1) \geq \frac{\gamma^2}{32}\left(d(\mu_t) - \frac{\eta}{2}\right)$$

The potential function Φ is bounded below by 0 and is initially equal to 1. The algorithm terminates if $d(\mu) < \eta$, so Lemma 14 implies that `Massart-Boost` terminates in $O(1/(\eta\gamma^2))$ rounds with high probability. A tighter analysis presented in Appendix C improves the round complexity to $O(\log^2(1/\eta)/\gamma^2)$ rounds.

Next, we show that a low density measure implies small misclassification error. Among $x \in \mathcal{X}^s$, the misclassification error is at most $d(\mu)$, since $\mu(x,y) = 1$ for any example such that $\operatorname{sign}(G(x)) \neq y$. To analyze the misclassification error on $x \in \mathcal{X}^r$, we proceed by casework on the rebalancing step being applied in the last round of boosting. We show that either i) $\mathcal{X}^r$ is at most an $\epsilon/2$-fraction of distribution D, or ii) the misclassification error on $\mathcal{X}^r$ is at most $\eta + \epsilon$. In the first case, we can assume $\operatorname{sign}(G)$ misclassifies every $x \in \mathcal{X}^r$ and still achieves error $\eta + \epsilon$. In the second case, the error on both $\mathcal{X}^s$ and $\mathcal{X}^r$ is at most $\eta + \epsilon$.

Lemma 15 (Label error) *When* `Massart-Boost` *terminates, with all but probability* $\delta/4$, *trained classifier* G *satisfies* $\mathrm{err}_{0\text{-}1}^{D}(\operatorname{sign}(G)) \leq \eta + \epsilon$, *assuming* $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$.

The following theorem combines these arguments with analyses of sample complexity, failure rate, and runtime to show that `Massart-Boost` efficiently converges to a low-error classifier.

Theorem 16 (Boosting Theorem) *Let* `WkL` *be an* (α, γ)*-weak learner requiring a sample of size* $m_{\texttt{WkL}}$*. Then for any* $\delta \in (0, 1/2]$*, any Massart distribution* D *with noise rate* $\eta < 1/2$*, and any* $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$*, taking* $\lambda = \gamma/8$ *and* $\kappa = \eta$*,* $\texttt{Massart-Boost}^{\texttt{WkL}}(\lambda, \kappa, \eta, \epsilon, \delta, \gamma, \alpha, m_{\texttt{WkL}})$ *will, with probability* $1 - \delta$*, run for* $T \in O\left(\log^2(1/\eta)/\gamma^2\right)$ *rounds, output a hypothesis* H *such that* $\mathrm{err}_{0\text{-}1}^{D}(H) \leq \eta + \epsilon$ *and* $\mathrm{err}_{0\text{-}1}^{D_x, f}(H) \leq \frac{\eta+\epsilon}{1-\eta}$*, make no more than*

$$m \in O\left(\frac{\log^2(1/\eta)}{\gamma^2}\left(\frac{\log(1/(\delta\eta\gamma))}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta} + \frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2}\right)\right)$$

calls to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$*, and run in time*

$$m \in O\left(\frac{\log^4(1/\eta)}{\gamma^4}\left(\frac{\log(1/(\delta\eta\gamma))}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta} + \frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2}\right)\right),$$

neglecting the runtime of the weak learner.

4. Lower Bound on Error for Massart Boosting

In this section, we show that no generic "black-box" boosting algorithm can achieve significantly better misclassification error than that of our algorithm, i.e., $\eta + \Theta(\eta\alpha)$, given a Massart noise oracle $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$. While this seemingly matches the lower bound for RCN boosting from (Kalai and Servedio, 2003), the RCN bound only implies a lower bound for the special case of Massart noise when $\eta = \mathrm{OPT}$. Moreover, that bound is not directly applicable in the Massart noise setting, since an RCN weak learner is not required to tolerate Massart noise.

We show that a lower bound of η still holds in the Massart noise setting even if OPT is much smaller than η. That is, boosting algorithms cannot be improved even when only a very small fraction of instances are actually noisy.

We consider the case where the target function $f \in \mathcal{C}$ is a pseudorandom function that is biased towards -1 labels. The noise function $\eta(x)$ is non-zero only on a small, random subset of preimages of -1 under f, and so there is a small fraction of examples $(x, 1)$ where it cannot be distinguished if $f(x) = 1$ and $\eta(x) = 0$ or if $f(x) = -1$ and $\eta(x) > 0$. If the distributions D_μ queried by the boosting algorithm never satisfy $\mathbb{E}_{(x,y)\sim D_\mu}[y] > -2\gamma$, an uncooperative weak learner can return the constant function -1 and have advantage γ. On the other hand, if the boosting algorithm does query the weak learner on a distribution such that $\mathbb{E}_{(x,y)\sim D_\mu}[y] > -2\gamma$, it risks overweighting a noisy example $(x, 1)$ and violating the Massart condition. Our adversarial, "rude" weak learner `rWkL` exploits this tension by only providing information attainable without knowing f.

4.1. Adversarial Weak Learner

Let `BlackBoxBoost` be a black-box boosting algorithm that draws m examples from EX and queries the weak learner T times. Let $\texttt{SP}_t$ be a sampling procedure defining $D^{\texttt{SP}_t}$, the t'th distribution queried to the weak learner. The following "rude" weak learner $\texttt{rWkL}_{m,T}(S)$ attempts to only give information that `BlackBoxBoost` could discover alone: `rWkL` identifies the heavy-hitters of $D^{\texttt{SP}_t}$ and defines $h(x)$ as x's majority label under $D^{\texttt{SP}_t}$. On non-heavy-hitters, $h(x)$ is the constant -1 function. For more details, see Appendix D.3. For pseudocode, see Algorithm 8.

4.2. Error Lower Bound Theorem

Theorem 17 (Error Lower Bound Theorem) *Let $\eta \in [0, 1/2), \alpha \in (0, 1/2 - \eta)$. Let $\{f_s\}$ be an η'-biased pseudorandom function family with security parameter n, where $\eta' = \eta(1 + \alpha/5)$. Let η, α be at least inversely polynomially in n bounded away from $1/2$. Then, for random s, no efficient black-box boosting algorithm* `BlackBoxBoost` *with example bound m running for T rounds, given query access to $(\alpha, \gamma(\alpha) \stackrel{\text{def}}{=} \alpha/20)$-weak learner $\texttt{rWkL}_{m,T}$ and $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ examples from example oracle $\mathrm{EX}(U_n, f_s, \eta(x))$, can output a hypothesis with label error at most $\eta(1 + o(\alpha))$. In particular, for all polynomials q, for all polynomial time black-box Massart boosting algorithms* `BlackBoxBoost` *with query access to* `rWkL` *and example oracle* EX*, for n sufficiently large, $\mathbf{Pr}_{s \in U_n}\left[\mathrm{err}_{\text{0-1}}^{U_n, f_s}(H) \le \eta'\right] < \frac{1}{q(n)}$, where H is the trained classifier output by* `BlackBoxBoost`.

A detailed proof is presented in Appendix D. Intuitively, since `rWkL` can be simulated without knowing f, `rWkL` cannot help the boosting algorithm learn f. So, no efficient algorithm can use `rWkL` to learn pseudorandom f. The primary focus of the proof is showing that `rWkL` is a valid Massart noise weak learner.

Adversarial weak learner `rWkL` has a stability property we call *reproducibility*. Assuming $D^{\mathtt{SP}_t}$ is Massart, `rWkL` returns the same exact hypothesis h^v with high probability over its sample $S \sim D^{\mathtt{SP}_t}$. Using reproducibility, we argue that i) boosting with `rWkL` can be efficiently simulated without knowing the function f and ii) `rWkL` satisfies the definition of a Massart noise weak learner. By Massart-ness, the labels of heavy hitters $x \in \mathcal{X}$ of $D^{\mathtt{SP}_t}$ must be biased towards the true label $f(x)$, guaranteeing advantage on heavy-hitters. To show `rWkL` also handles non-heavy-hitters gracefully, we first show that boosting with `rWkL` can be efficiently simulated, and then we appeal to the pseudorandomness of f. `rWkL` only fails to return a hypothesis with γ advantage if $D^{\mathtt{SP}_t}$ is supported on many non-heavy-hitters x whose true label $f(x) = 1$. We can conclude by observing that finding many such non-heavy-hitters implies a violation of the pseudorandomness assumption.

Lemma 18 (Advantage of `rWkL`) *Let $D^{\mathtt{SP}_t}$ denote the distribution induced by the sampling procedure $\mathtt{SP}_t$ and* `hEG` *at round $t \in [T]$ of boosting. Similarly, let $D_r^{\mathtt{SP}_t}$ denote the distribution induced by $\mathtt{SP}_t$ and* `rEG`*. Let S_t denote a sample drawn i.i.d. from $D_r^{\mathtt{SP}_t}$. Then for all $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting* `rWkL` *with* `rEG`*, if $D^{\mathtt{SP}_t}$ is Massart, then with probability $1 - O(1/(mT))$ over its internal randomness, $\texttt{rWkL}(S_t)$ outputs a hypothesis h_t with advantage at least γ against $D^{\mathtt{SP}_t}$, except with negligible probability in m over the choice of $\mathtt{SP}_t$.*

Acknowledgments

Supported by NSF Award CCF-1652862 (CAREER), a Sloan Research Fellowship, and a DARPA Learning with Less Labels (LwLL) grant. Supported by the Simons Foundation and NSF grant CCF-1909634. Supported by NSF Award CCF-1553288 (CAREER) and a Sloan Research Fellowship. Supported by NSF grant CCF-2008006.

References

D. Angluin and P. Laird. Learning from noisy examples. *Mach. Learn.*, 2(4):343–370, 1988.

P. Awasthi, M. F. Balcan, N. Haghtalab, and R. Urner. Efficient learning of linear separators under bounded noise. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015*, pages 167–190, 2015.

P. Awasthi, M. F. Balcan, N. Haghtalab, and H. Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016*, pages 152–192, 2016.

B. Barak, M. Hardt, and S. Kale. The uniform hardcore lemma via approximate bregman projections. In Claire Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009*, pages 1193–1200. SIAM, 2009.

E. Beigman and B. B. Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, 2009.

S. Ben-David, P. Long, and Y. Mansour. Agnostic boosting. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, pages 507–516, 2001.

A. Blum. Machine learning: My favorite results, directions, and open problems. In *44th Symposium on Foundations of Computer Science (FOCS 2003)*, pages 11–14, 2003.

J. Bradley and R. Schapire. Filterboost: Regression and classification on large datasets. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.

P. Bühlmann and T. Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statist. Sci.*, 22(4):477–505, Nov 2007.

S. Chen, F. Koehler, A. Moitra, and M. Yau. Classification under misspecification: Halfspaces, generalized linear models, and connections to evolvability. *CoRR*, abs/2006.04787, 2020.

E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings of the Thirty-Eighth Symposium on Foundations of Computer Science*, pages 514–521, 1997.

A. Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the 48th Annual Symposium on Theory of Computing, STOC 2016*, pages 105–117, 2016.

I. Diakonikolas and D. M. Kane. Hardness of learning halfspaces with massart noise. *CoRR*, abs/2012.09720, 2020.

I. Diakonikolas, T. Gouleakis, and C. Tzamos. Distribution-independent PAC learning of halfspaces with massart noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 4751–4762, 2019.

I. Diakonikolas, D. Kane, V. Kontonis, C. Tzamos, and N. Zarifis. A polynomial time algorithm for learning halfspaces with tsybakov noise. *CoRR*, abs/2010.01705, 2020a.

I. Diakonikolas, V. Kontonis, C. Tzamos, and N. Zarifis. Learning halfspaces with massart noise under structured distributions. In *Conference on Learning Theory, COLT 2020*, volume 125 of *Proceedings of Machine Learning Research*, pages 1486–1513. PMLR, 2020b.

I. Diakonikolas, V. Kontonis, C. Tzamos, and N. Zarifis. Learning halfspaces with tsybakov noise. *CoRR*, abs/2006.06467, 2020c.

V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science*, pages 241–250, 2010.

V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu. Agnostic learning of monomials by halfspaces is hard. In *FOCS*, pages 385–394, 2009.

B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2): 256–285, 1995.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997a.

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

D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *JMLR*, 4: 101–117, 2003.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

T. Holenstein. Key agreement from weak bit agreement. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 664–673. ACM, 2005.

R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the Thirty-Sixth Annual Symposium on Foundations of Computer Science*, pages 538–545, 1995.

A. Kalai and V. Kanade. Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 880–888, 2009.

A. Kalai and R. Servedio. Boosting in the presence of noise. In *Proceedings of the 35th Annual Symposium on Theory of Computing (STOC)*, pages 196–205, 2003.

A. Kalai, Y. Mansour, and E. Verbin. On agnostic boosting and parity learning. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 629–638, 2008.

M. Kearns, R. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17 (2/3):115–141, 1994.

M. J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6): 983–1006, 1998.

A. R. Klivans and R. A. Servedio. Boosting and hard-core sets. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99*, pages 624–633. IEEE Computer Society, 1999.

P. Long and R. Servedio. Martingale boosting. In *Proc. 18th Annual Conference on Learning Theory (COLT)*, pages 79–94, 2005.

P. Long and R. Servedio. Adaptive martingale boosting. In *Proc. 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 977–984, 2008.

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

Y. Mansour and D. McAllester. Boosting using branching programs. *Journal of Computer & System Sciences*, 64(1):103–112, 2002.

P. Massart and E. Nedelec. Risk bounds for statistical learning. *Ann. Statist.*, 34(5):2326–2366, 10 2006.

R. Rivest and R. Sloan. A formal model of hierarchical concept learning. *Information and Computation*, 114(1):88–114, 1994.

R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

R. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, and B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.

R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012. ISBN 0262017180.

R. Servedio. Smooth boosting and learning with malicious noise. *JMLR*, 4:633–648, 2003.

R. H. Sloan. Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, COLT '88, pages 91–96, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

R. H. Sloan. Corrigendum to types of noise in data for concept learning. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992*, page 450, 1992.

R. H. Sloan. *Pac Learning, Noise, and Geometry*, pages 21–41. Birkhäuser Boston, Boston, MA, 1996.

L. G. Valiant. A theory of the learnable. In *Proc. 16th Annual ACM Symposium on Theory of Computing (STOC)*, pages 436–445. ACM Press, 1984.

V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics*. Springer-Verlag, Berlin, Heidelberg, 1982. ISBN 0387907335.

C. Zhang, J. Shen, and P. Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. *CoRR*, abs/2002.04840, 2020.

Y. Zhang, P. Liang, and M. Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*, pages 1980–2022, 2017.

Appendix A. Additional Definitions and Proofs for Section 2

A.1. Boosting and Weak Learners

Recall the definition of a Massart noise weak learner.

Definition 19 (Massart Noise Weak Learner) *Let $\mathcal{C}$ be a concept class of functions $f : \mathcal{X} \to \{\pm 1\}$. Let $\alpha \in [0, 1/2)$. Let $\gamma : \mathbb{R} \to \mathbb{R}$ be a function of α. A Massart noise (α, γ)-weak learner* `WkL` *for $\mathcal{C}$ is an algorithm such that, for any distribution D_x over $\mathcal{X}$, function $f \in \mathcal{C}$, and noise function $\eta(x)$ with noise bound $\eta < 1/2 - \alpha$,* `WkL` *outputs a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ such that $\mathbf{Pr}_S[\mathrm{adv}^D(h) \geq \gamma] \geq 2/3$, where the sample S is drawn from Massart distribution $D = \mathtt{Mas}\{f, D_x, \eta(x)\}$.*

Note that we define an (α, γ)-Massart noise weak learner to have failure probability at most $1/3$. For any desired $\delta \in (0, 1/3)$, such a weak learner can be used to obtain a hypothesis with advantage $\gamma/2$, with all but probability δ, by standard repetition techniques demonstrated in Lemma 20.

Lemma 20 (`WkL` repetition) *Let* `WkL` *be an (α, γ)-Massart noise weak learner requiring a sample of size $m_{\mathtt{WkL}}$. Then for any $\delta \in (0, 1/3)$, $2\log(2/\delta)$ calls to* `WkL` *and $2\log(2/\delta)(m_{\mathtt{WkL}} + 1/\gamma^2)$ examples suffice to obtain a hypothesis with advantage at least $\gamma/2$ with all but probability δ.*

Proof To drive down the failure probability of `WkL`, we draw $2\log(2/\delta)$ samples of size $m_{\mathtt{WkL}}$ and run `WkL` on each of them to obtain a list of hypotheses, at least one of which has advantage γ with all but probability $\delta/2$. We then draw a sample of size $2\log(2/\delta)/\gamma^2$ to test each hypothesis in our list, keeping the best. The Chernoff-Hoeffding inequality guarantees that testing our hypotheses overestimates the advantage by more than $\gamma/2$ with probability no greater than $\delta/2$, and so we obtain a hypothesis with advantage at least $\gamma/2$ with all but probability δ. ■

We are primarily interested in *efficient* Massart noise weak learners (Definition 21).

Definition 21 (Efficient Massart Noise Weak Learner) *Let* `WkL` *be an (α, γ)-Massart noise weak learner. Let n be the maximum bit complexity of a single example $(x, y) \in \mathcal{X} \times \{\pm 1\}$, and let $m_{\mathtt{WkL}}$ denote the number of examples comprising sample S.* `WkL`(S) *is* efficient *if*

1. `WkL` *uses $m_{\mathtt{WkL}}(n, \eta, \gamma) = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ examples.*
2. `WkL` *outputs a hypothesis h in time* $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.
3. *Hypothesis $h(x)$ has bit complexity* $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.
4. *For all $x \in \mathcal{X}$, the hypothesis $h(x)$ can be evaluated in time* $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.

Boosting algorithms can utilize the advantage guarantee of the weak learner by cleverly reweighting its input distributions. To sample from these reweighted distributions, we sample from the underlying distribution D via $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ and reject examples according to a function $\mu : \mathcal{X} \times \{\pm 1\} \to [0, 1]$. We refer to μ informally as a *measure* to emphasize that it induces a distribution, but need not be one itself.

Definition 22 (Rejection Sampled Distribution D_μ) *Let D be a Massart distribution, and let $\mu : \mathcal{X} \times \{\pm 1\} \to [0,1]$ be an efficiently computable measure. We define D_μ as the distribution generated from D by the following rejection sampling procedure: draw an example $(x, y) \sim D$. With probability $\mu(x, y)$, keep this example. Otherwise, repeat this process (until an example is kept).*

Note that some choices of μ may induce a distribution D_μ which is *not* Massart, as reweighting examples may distort $\eta(x)$, and so it is possible that we no longer have a noise bound less than $1/2$. In particular, if there is an $x \in \mathrm{supp}(D_x)$ for which $\mu(x, -f(x))\eta(x) \gg \mu(x, f(x))(1-\eta(x))$, then D_μ is not a Massart distribution and running the weak learner on a sample from this distribution is not guaranteed to return a hypothesis with good advantage. In designing our boosting algorithm, we will choose μ carefully to ensure that this never happens.

Lemma 23 (Sampling from D_μ) *For any $m > 0$, $\delta \in (0, 1/2)$, obtaining a sample of size m from D_μ by rejection sampling from D requires no more than $\frac{\log(1/\delta)}{d(\mu)^2} + \frac{2m}{d(\mu)}$ examples from distribution D, with all but probability δ.*

Proof From the definition of D_μ, we can sample from D_μ by drawing an example (x, y) from D and keeping it with probability $\mu(x, y)$. By Definition 11, we expect to keep an example with probability $d(\mu)$. Then the Chernoff-Hoeffding inequality allows us to conclude that, following this procedure, if we draw $\frac{\log(1/\delta)}{d(\mu)^2} + \frac{2m}{d(\mu)}$ examples from D, we keep at least m of them with all but probability δ. ■

Appendix B. Additional Lemmas and Proofs for Section 3

B.1. Boosting Algorithm

In this appendix, we present our Massart noise-tolerant boosting algorithm `Massart-Boost` (Algorithm 2) and prove the following theorem:

Theorem 24 ((Simplified) Boosting Theorem) *Let `WkL` be an (α, γ)-weak learner requiring a sample of size $m_{\texttt{WkL}}$. Then for any Massart distribution D with noise rate $\eta < 1/2$, and any $\epsilon > \frac{8\eta\alpha}{1-2\alpha}$, `Massart-Boost` will*

- *make at most $T \in \tilde{O}\left(1/(\eta\gamma^2)\right)$ calls to `WkL`*
- *output a hypothesis H such that $\mathrm{err}_{\text{0-1}}^{D}(H) \leq \eta + \epsilon$ and $\mathrm{err}_{\text{0-1}}^{D_x, f}(H) \leq \frac{\eta+\epsilon}{1-\eta}$*
- *make*
$$m \in \tilde{O}\left(\frac{1}{\eta\gamma^2\epsilon^3} + \frac{m_{\texttt{WkL}}}{\eta^2\gamma^2} + \frac{1}{\eta^2\gamma^4}\right)$$
calls to its example oracle $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$
- *run in time*
$$\tilde{O}\left(\frac{m_{\texttt{WkL}}}{\eta^3\gamma^4} + \frac{1}{\eta^3\gamma^6} + \frac{1}{\eta^2\gamma^4\epsilon^3}\right),$$
neglecting the runtime of the weak learner.

B.2. Description of Boosting Algorithm

The pseudocode for our boosting algorithm is Algorithm 2. It makes calls to three subroutines in addition to the weak learner: `Samp` (Routine 3), `Est-Density` (Routine 4), and `OverConfident` (Routine 5), which we first describe informally.

The `Samp` subroutine captures the procedure by which our algorithm draws samples for the weak learner. These samples are drawn i.i.d. from the reweighted distributions constructed by our booster. The `Samp` procedure is given oracle access to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, so that is can sample from D. The `Samp` procedure takes as input a function (the current hypothesis) G, the size of the sample $m_{\texttt{WkL}}$ required by the weak learner, and the threshold s for $|G(x)|$ that defines which examples are to be withheld from the weak learner. `Samp` repeatedly draws examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, and keeps them with probability $\mu_{G,s}(x, y)$, the value of which is computed using G and s. After `Samp` has drawn a sample of size $m_{\texttt{WkL}}$, it returns the sample, and this is what is given to the weak learner as input.

The subroutine `Est-Density` is used to estimate the current density of the measure μ_t, which is necessary to test the termination condition of our algorithm. Ideally, the algorithm terminates once $d(\mu_t) < \kappa$, and so `Est-Density` is called at the end of each round of boosting to estimate this density. `Est-Density` is given oracle access to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ and takes as input G and s, so that it can empirically estimate $d(\mu)$ using a sample drawn from D. `Est-Density` also takes as input three parameters: $\delta_{\texttt{dens}}$, ϵ, and η. The parameters ϵ and η are used to specify the desired accuracy of the density estimation, $\beta = \min\{\epsilon/2, \eta/4\}$. The parameter $\delta_{\texttt{dens}}$ specifies the tolerable probability of failure of the density estimation procedure (i.e., the probability that `Est-Density` returns an estimate of $d(\mu)$ with error greater than β).

The subroutine `OverConfident` determines when the error of $\mathrm{sign}(G)$ on examples withheld from the weak learner (i.e., examples in $\mathcal{X}_G^r$) has grown too large. If, at round t, the probability mass on $\mathcal{X}_t^r$ is large, and the error of $\mathrm{sign}(G_t)$ on $\mathcal{X}_t^r$ exceeds $\eta + \epsilon$, we must improve G_t on these examples to reach our target error of $\eta + \epsilon$. Because we withhold examples in $\mathcal{X}_t^r$ from the weak learner at round $t + 1$, we are not guaranteed that the next weak hypothesis, h_{t+1}, will provide any amount of progress in expectation on these examples, so some additional steps are needed to improve G_t. The role of `OverConfident` is to estimate whether the probability mass of $\mathcal{X}_t^r$ is significant, and if so, whether the error of $\mathrm{sign}(G_t)$ on $\mathcal{X}_t^r$ is large enough that an additional correction step is necessary.

The subroutine `OverConfident` is given oracle access to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ and takes as input G, s, η, and ϵ. It also takes an additional parameter $\delta_{\texttt{err}}$, which specifies the tolerable probability of failure for `OverConfident` (i.e., the probability that `OverConfident` returns a false positive or false negative). `OverConfident` first estimates the probability that $|G(x)| > s$. If it estimates $\mathbf{Pr}_{(x,y)\sim D}[|G(x)| \geq s] < \epsilon/4$, then the overall contribution of examples in $\mathcal{X}_G^r$ to the total error of $\mathrm{sign}(G)$ is sufficiently small that the correction step is not needed. In this case, `OverConfident` returns false. If it estimates the probability to be greater than $\epsilon/4$, it draws a new sample for estimating the conditional error of G on $\mathcal{X}_G^r$. The subroutine makes calls to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ and keeps only the examples (x, y) such that $x \in \mathcal{X}_G^r$. It draws a sufficiently large sample to estimate the error of G on this set to within $\epsilon/4$ with all but probability $\delta_{\texttt{err}}/2$. If the estimated error exceeds $\eta + 3\epsilon/4$, it returns true and the correction step takes place. Otherwise it returns false, as the conditional error on these points is tolerable.

Algorithm 2 $\texttt{Massart-Boost}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x)),\texttt{WkL}}(\lambda,\kappa,\eta,\epsilon,\delta,\gamma,\alpha,m_{\texttt{WkL}})$
λ: Learning rate
κ: Target density for measure μ
η: Massart noise rate
ϵ: Target error in excess of η
δ: Target failure probability for `Massart-Boost`
γ: Weak learner advantage guarantee
α: Weak learner parameter indicating `WkL` can tolerate noise rate $\eta < 1/2 - \alpha$
$m_{\texttt{WkL}}$: Sample size for `WkL`

$c \leftarrow \frac{4\eta\alpha}{1-2\alpha}$, $s \leftarrow \log\left(\frac{1-\eta}{\eta+c}\right)$ // set parameters for managing noise
$\delta_{\texttt{err}} \leftarrow \delta\eta\gamma^2/1536$, $\delta_{\texttt{dens}} \leftarrow \delta\eta\gamma^2/1024$ // set failure probabilities for subroutines
$G_0 \leftarrow 0$, $t \leftarrow 0$ // initialize G, round counter
$\widehat{d} \leftarrow 1$ // initialize density estimate
while $\widehat{d} > \kappa$ **do**
  $t \leftarrow t+1$
  $S \leftarrow \texttt{Samp}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_{t-1},n,s)$ // draw a sample for the weak learner
  $h_t \leftarrow \texttt{WkL}(S)$ // obtain a weak hypothesis
  $h_t^s(x) \leftarrow \begin{cases} h_t(x) & \text{if } x \in \mathcal{X}_{t-1}^s, \\ 0 & \text{otherwise} \end{cases}$ // zero out hypothesis on $\mathcal{X}_t^r$
  $G_t \leftarrow G_{t-1} + \lambda h_t^s$ // update working hypothesis
  if $\texttt{OverConfident}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_t,s,\delta_{\texttt{err}},\epsilon)$ **then**
    $h_t^r(x) \leftarrow \begin{cases} -\mathrm{sign}(G_t(x)) & \text{if } x \in \mathcal{X}_t^r, \\ 0 & \text{otherwise} \end{cases}$ // if error on $\mathcal{X}_t^r$ is high, be less confident
    $G_t \leftarrow G_t + \lambda h_t^r$ // update working hypothesis
  $\widehat{d} \leftarrow \texttt{Est-Density}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_t,s,\delta_{\texttt{dens}},\epsilon)$// estimate density of measure
$H \leftarrow \mathrm{sign}(G_t)$
return H

B.3. Convergence of `Massart-Boost`

In this subsection, we bound the error of the final hypothesis output by Algorithm 2 and the number of rounds of boosting required to achieve this error bound. We begin by showing an invariant of our algorithm that will be useful in subsequent potential arguments.

Lemma 25 (Invariant for $|G_t(x)|$) *For all rounds t of boosting and all examples $(x,y) \in (\mathcal{X},\mathcal{Y})$, at the end of round t, $|G_t(x)| < s+\lambda$.*

Proof We first show that at the end of round t, $|G_t(x)| \leq s+\lambda$. On examples x such that $|G_{t-1}(x)| \geq s$, either $G_t(x) = G_{t-1}(x) - \lambda\mathrm{sign}(G_{t-1}(x))$ or $G_t(x) = G_{t-1}(x)$, and so $|G_t(x)| \leq |G_{t-1}(x)|$. Since $|G_t(x)| \geq |G_{t-1}(x)|$ only when $|G_{t-1}(x)| < s$, we now consider how much larger it can be. For examples such that $|G_{t-1}(x)| < s$, either $G_t(x) = G_{t-1}(x) + \lambda h_t(x) + \lambda h_t^r(x)$

Routine 3 $\texttt{Samp}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G, m_{\texttt{WkL}}, s)$

$S \leftarrow \emptyset$
while $|S| \leq m_{\texttt{WkL}}$ **do**
  $(x, y) \leftarrow \mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$
  With prob $\mu_{G,s}(x, y)$, $S \leftarrow S \,||\, (x, y)$ // draw a sample for WkL from D_μ
return S

Routine 4 $\texttt{Est-Density}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G, s, \delta_{\texttt{dens}}, \epsilon)$

$\beta \leftarrow \min\{\epsilon/2, \eta/4\}$
Draw set S of $\log(1/\delta_{\texttt{dens}})/(2\beta^2)$ examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$
$\widehat{d} \leftarrow \frac{1}{|S|} \sum_{(x,y)\in S} \mu_{G,s}(x, y)$ // estimate the density of μ
return $\widehat{d}$

Routine 5 $\texttt{OverConfident}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G, s, \delta_{\texttt{err}}, \epsilon)$

Draw set S of $32\log(2/\delta_{\texttt{err}})/\epsilon^2$ examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$
if $\frac{|S\cap\mathcal{X}_G^r|}{|S|} \leq \epsilon/4$ **then**
  return false // if $\mathcal{X}_G^r$ is small, return false
$S \leftarrow \emptyset$
while $|S| \leq 8\log(2/\delta_{\texttt{err}})/\epsilon^2$ **do**
  $(x, y) \leftarrow \mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$
  if $|G(x)| \geq s$ **then**
    $S \leftarrow S \,||\, (x, y)$
$\widehat{\epsilon} \leftarrow \frac{1}{2|S|} \sum_{(x,y)\in S} |y - \mathrm{sign}(G(x))|$ // if $\mathcal{X}_G^r$ is large, estimate error on $\mathcal{X}_G^r$
if $\widehat{\epsilon} \geq \eta + 3\epsilon/4$ **then**
  return true // if error and $\mathcal{X}_G^r$ are large, return true
else
  return false // if error is small, return false

(when `OverConfident` returns true) or $G_t(x) = G_{t-1}(x) + \lambda h_t(x)$. In the first case, if $\lambda h_t^r(x) \neq 0$, then $\operatorname{sign}(G_{t-1}(x) + h_t(x)) = -h_t^r(x)$, and so $|G_t(x)| \leq |G_{t-1}(x) + \lambda h_t(x)|$ for both cases. Since the hypothesis $h_t(x)$ output by the weak learner has codomain $[-1, 1]$, it follows that $|G_t(x)| \leq |G_{t-1}(x) + \lambda h_t(x)| < s + \lambda$. ■

We now recall the potential function introduced in Section 3.2. We denote by $\phi_t(x, y)$ the function

$$\phi_t(x, y) = \int_{yG_t(x)}^{\infty} M(z) dz.$$

Then our potential function is defined as

$$\Phi(t) = \mathop{\mathbb{E}}_{(x,y)\sim D}[\phi_t(x, y)] = \mathop{\mathbb{E}}_{(x,y)\sim D} \int_{yG_t(x)}^{\infty} M(z) dz.$$

We will make use of the following upper-bound on $d(\mu_t)$ in terms of $\Phi(t)$.

Lemma 26 (Potential upper-bounds density) *For every round t of* `Massart-Boost`*, $d(\mu_t) \leq \Phi(t)$.*

Proof We show that $d(\mu_t) \leq \Phi_t$ by showing $\mu_t(x, y) \leq \phi_t(x, y)$. For examples (x, y) such that $yG_t(x) > 0$, we have

$$\phi_t(x, y) = \int_{yG_t(x)}^{\infty} e^{-z} dz = e^{-yG_t(x)} \geq \mu_t(x, y).$$

For the remaining points, we simply observe that either $\mu_t(x, y) = 1$ or $\mu_t(x, y) = 0$. In either case, the potential

$$\phi_t(x, y) = \int_{yG_t(x)}^{\infty} M(z) dz \geq \int_{0}^{\infty} e^{-z} dz = 1$$

and so we have that $\mu_t(x, y) \leq \phi_t(x, y)$, and therefore $d(\mu_t) \leq \Phi_t$. ■

We now prove that `Massart-Boost` makes progress against Φ at each round.

Lemma 27 (Potential Drop) *Take $\lambda = \gamma/8$, $\delta_{\texttt{WkL}} = \delta\eta\gamma^2/1536$, and assume $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$. Then for every round of boosting t, with all but probability $\delta\eta\gamma^2/768$,*

$$\Phi(t) - \Phi(t+1) \geq \frac{\gamma^2}{32} \left(d(\mu_t) - \frac{\eta}{2} \right)$$

Proof We first show that for all $(x, y) \sim D$ such that $x \in \mathcal{X}_t^s$,

$$\phi_t(x, y) - \phi_{t+1}(x, y) \geq \lambda \mu_t(x, y)(y h_t(x) - 2\lambda).$$

We prove this statement for examples such that $0 \leq yG_t(x), yG_{t+1}(x) < s$, i.e., in the non-constant region of M, and observe that this suffices to prove the statement for all (x, y) such that

$x \in \mathcal{X}_t^s$. To see that this is true, note that within the constant region of M, $\phi_t(x,y) - \phi_{t+1}(x,y) = \lambda\mu_t(x,y)yh_t(x)$. For examples moved by h_t from constant to non-constant regions of M,

$$\begin{aligned}
\phi_t(x,y) - \phi_{t+1}(x,y) &= \int_{yG_t(x)}^{0} 1dz + \int_{0}^{yG_{t+1}(x)} e^{-z}dz \\
&\geq \int_{0}^{\lambda y h_t(x)} e^{-z}dz \\
&\geq \lambda M(0)(yh_t(x) - 2\lambda) && \text{(by assumption)} \\
&= \lambda\mu_t(x,y)(yh_t(x) - 2\lambda)
\end{aligned}$$

Similarly, for examples moving into the constant region from non-constant,

$$\begin{aligned}
\phi_t(x,y) - \phi_{t+1}(x,y) &= -\int_{yG_{t+1}(x)}^{0} 1dz - \int_{0}^{yG_t(x)} e^{-z}dz \\
&= \int_{yG_t(x)}^{0} e^{-z}dz + \int_{0}^{yG_{t+1}(x)} 1dz \\
&\geq \int_{yG_t(x)}^{yG_{t+1}} e^{-z}dz \\
&\geq \lambda\mu_t(x,y)(yh_t(x) - 2\lambda) && \text{(by assumption/proved below)}
\end{aligned}$$

and so it only remains to prove the claim for examples such that $0 \leq yG_t(x), yG_{t+1}(x) < s$. By the definition of ϕ_t, we have

$$\begin{aligned}
\phi_t(x,y) - \phi_{t+1}(x,y) &= \int_{yG_t(x)}^{yG_{t+1}(x)} M(z)dz \\
&= \int_{yG_t(x)}^{yG_{t+1}(x)} e^{-z}dz \\
&= e^{-v}(yG_{t+1}(x) - yG_t(x)) && \text{(for some } v \in [yG_t(x), yG_{t+1}(x)]) \\
&\geq e^{-yG_{t+1}(x)}\lambda y h_t(x) \\
&= e^{-yG_t(x)}e^{-\lambda y h_t(x)}\lambda y h_t(x) \\
&\geq \mu_t(x,y)\lambda y h_t(x) - 2\mu_t(x,y)\lambda^2 && ((xe^{-x} \geq x - 2x^2 \text{ for } x \in [-1,1]) \\
&= \lambda\mu_t(x,y)(yh_t(x) - 2\lambda)
\end{aligned}$$

and so the contribution to the potential drop from $(x,y) \in \mathcal{X}_t^s$ is as claimed.

We now consider the contribution to the potential drop from examples (x,y) where $x \in \mathcal{X}_t^r$, by analyzing two complementary cases.

1. $\texttt{OverConfident}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_t, s, \delta_{\texttt{err}}, \epsilon)$ returns false

2. $\texttt{OverConfident}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_t, s, \delta_{\texttt{err}}, \epsilon)$ returns true

In the first case, $h_t^r = 0$, and so $yG_{t+1}(x) = yG_t(x)$ holds for all these examples. Therefore the contribution to the potential drop is

$$\mathop{\mathbb{E}}_{(x,y)\sim D}\left[\phi_t(x,y) - \phi_{t+1}(x,y)\middle| x \in \mathcal{X}_t^r\right] = 0.$$

In the second case, $h_t^r(x) = -\mathrm{sign}(G_t(x))$, and so $yG_{t+1}(x) = yG_t(x) - \mathrm{sign}(G_t(x))$ for these examples. $\texttt{OverConfident}^{\mathrm{EX}^{\mathrm{Mas}}(f,D_x,\eta(x))}(G_t, s, \delta_{\texttt{err}}, \epsilon)$ only returns true if it has estimated the error on examples such that $x \in \mathcal{X}_t^r$ exceeds $\eta + 3\epsilon/4$. This routine estimates the error from a sample of size $8\log(2/\delta_{\texttt{err}})/\epsilon^2$, and so it holds by the Chernoff-Hoeffding inequality that with all but probability $\delta_{\texttt{err}}/2$, that

$$\mathbf{Pr}_{(x,y)\sim D}\left[yG_t(x) \le -s\middle| x \in \mathcal{X}_t^r\right] \ge \eta + \epsilon/2.$$

This implies a contribution to the potential drop of

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim D}\left[\phi_t(x,y) - \phi_{t+1}(x,y)\middle| x \in \mathcal{X}_t^r\right] &= \mathbf{Pr}_{(x,y)\sim D}[yG_t(x) \le -s\big| x \in \mathcal{X}_t^r] \int_{yG_t(x)}^{yG_t(x)+\lambda} 1dz \\
&\quad + \mathbf{Pr}_{(x,y)\sim D}[yG_t(x) \ge s\big| x \in \mathcal{X}_t^r] \int_{yG_t(x)}^{yG_t(x)-\lambda} e^{-z}dz \\
&\ge (\eta+\epsilon/2)\lambda + (1-\eta-\epsilon/2)\int_{yG_t(x)}^{yG_t(x)-\lambda} e^{-z}dz \\
&\ge (\eta+\epsilon/2)\lambda + (1-\eta-\epsilon/2)e^{-s}(1-e^{-\lambda}) \\
&\qquad \text{(from } yG_t(x) \le s+\lambda\text{)} \\
&\ge (\eta+\epsilon/2)\lambda - (1-\eta-\epsilon/2)e^{-s}(\lambda-\lambda^2) \\
&\qquad \text{(from } e^{-\lambda} \le 1-\lambda+\lambda^2\text{)} \\
&= (\eta+\epsilon/2)\lambda - (1-\eta-\epsilon/2)(\frac{\eta+c}{1-\eta})(\lambda-\lambda^2) \\
&\qquad \text{(by definition of } s\text{)} \\
&\ge \frac{\epsilon\lambda}{2}(1+\eta-\eta\lambda) - c\lambda(1-\lambda) - \eta\lambda^2, \\
&\qquad \text{(from } \eta < \tfrac{\eta+c}{1-\eta}\text{)}
\end{aligned}$$

and so as long as $c \le \epsilon/2 \le \frac{\epsilon(1+\eta-\eta\lambda)}{2(1-\lambda)}$, we have

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[\phi_t(x,y) - \phi_{t+1}(x,y)\big| x \in \mathcal{X}_t^r] \ge -\eta\lambda^2.$$

Recall that we have assumed $\epsilon \ge \frac{8\eta\alpha}{1-2\alpha} = 2c$ and so the stated bound holds.

We now lower-bound the drop in the potential function. With probability $1-\delta_{\texttt{err}}$, $\texttt{Est-Density}$ does not overestimate the error of the current hypothesis on points x for which $x \in \mathcal{X}_t^r$ by more than $\epsilon/4$, and so we have

$$\begin{aligned}
\Phi(t)-\Phi(t+1) &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\phi_t(x,y)-\phi_{t+1}(x,y)] \\
&\geq \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^s] \mathop{\mathbb{E}}_{(x,y)\sim D}[\lambda\mu_t(x,y)(yh_t(x)-2\lambda)\big|x\in\mathcal{X}_t^s] \\
&\qquad - \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^r]\eta\lambda^2 \\
&= \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^s] \mathop{\mathbb{E}}_{(x,y)\sim D}[\lambda\mu_t(x,y)(yh_t(x)-2\lambda)\big|x\in\mathcal{X}_t^s] \\
&\qquad - \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^r]\eta\lambda^2 \\
&\qquad + \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^r] \mathop{\mathbb{E}}_{(x,y)\sim D}[\lambda\mu_t(x,y)(yh_t(x)-2\lambda)\big|x\in\mathcal{X}_t^r] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(since } \mu_t(x,y)=0 \text{ for } x\in\mathcal{X}_t^r\text{)} \\
&\geq \mathop{\mathbb{E}}_{(x,y)\sim D}[\lambda\mu_t(x,y)(yh_t(x)-2\lambda)] - \mathbf{Pr}_{(x,y)\sim D}[x\in\mathcal{X}_t^r](\eta\lambda^2) \\
&\geq \mathop{\mathbb{E}}_{(x,y)\sim D}[\lambda\mu_t(x,y)(yh_t(x)-2\lambda)] - \eta\lambda^2.
\end{aligned}$$

From our weak learner guarantee and Lemma 20, we know that with all but probability $\delta_{\mathtt{WkL}}$, h_t has advantage $\gamma/2$ against D_{μ_t}. Therefore with all but probability $\delta_{\mathtt{WkL}}+\delta_{\mathtt{err}}$,

$$\Phi(t)-\Phi(t+1)\geq\frac{\lambda\gamma}{2}d(\mu_t)-2\lambda^2 d(\mu_t)-\eta\lambda^2.$$

Then taking $\delta_{\mathtt{WkL}}=\delta_{\mathtt{err}}=\frac{\delta\eta\gamma^2}{1536}$ and $\lambda=\gamma/8$, we have

$$\Phi(t)-\Phi(t+1)\geq\frac{\gamma^2}{32}\left(d(\mu_t)-\frac{\eta}{2}\right)$$

with all but probability $\delta\eta\gamma^2/768$. ■

Now we use our guaranteed drop in potential to show bounds on termination, as well as the density of the measure μ_t at the end of the final round t.

Lemma 28 (Termination) *Let* `WkL` *be an* (α,γ)*-weak learner requiring a sample of size* $m_{\mathtt{WkL}}$ *and let* $\delta_{\mathtt{WkL}}=\delta\eta\gamma^2/1536$. *Let* $\lambda=\gamma/8$ *and* $\kappa\geq\eta$. *Then with all but probability* $\delta/3$, `Massart-Boost`$^{\mathtt{WkL}}$ *terminates within* $T\leq 128/(\eta\gamma^2)$ *rounds, and conditioned on termination,* $d(\mu_T)\leq\kappa+\epsilon/2$ *with all but probability* $\delta/8$.

Proof `Massart-Boost` terminates once `Est-Density` estimates $\widehat{d}(\mu_t)\leq\kappa$. Given that `Est-Density` draws a sample of size $2\log(1/\delta_{\mathtt{dens}})/\beta^2$ for $\beta=\min\{\epsilon/2,\eta/4\}$, the Chernoff-Hoeffding inequality bounds the probability that `Est-Density` overestimates $d(\mu_t)$ by more than β by $\delta_{\mathtt{dens}}$. Therefore the probability that `Massart-Boost` fails to terminate at the end of any round for which $d(\mu_t)\leq\kappa-\beta$ is no more than $\delta_{\mathtt{dens}}$. We condition on this failure not occuring for the rest of the proof.

From Lemma 14, we have that with probability at least $1-\delta_{\mathtt{WkL}}-\delta_{\mathtt{err}}$,

$$\Phi(t)-\Phi(t+1)\geq\frac{\gamma^2}{32}\left(d(\mu_t)-\frac{\eta}{2}\right).$$

We have taken $\kappa \geq \eta$, and $\beta \leq \eta/4$, so except with probability $\delta_{\texttt{WkL}} + \delta_{\texttt{err}}$, the potential drops by at least $\frac{\gamma^2}{32}(\kappa - \beta - \frac{\eta}{2}) > \frac{\eta\gamma^2}{128}$ in each round. The potential function begins at

$$\Phi_0 = \mathop{\mathbb{E}}_{(x,y)\sim D} \int_0^\infty M(z)dz = 1$$

and has minimum value 0, so taking $T = \frac{128}{\eta\gamma^2}$, it must be the case that $d(\mu) \leq \kappa - \beta$ by round T with probability at least $1 - T(\delta_{\texttt{WkL}} + \delta_{\texttt{err}})$. So with all but probability $128(\delta_{\texttt{WkL}} + \delta_{\texttt{err}})/(\eta\gamma^2)$, $d(\mu) \leq \kappa - \beta$ after T rounds, and so `Massart-Boost` must have terminated by then except with probability $\delta_{\texttt{dens}}$. This gives a total failure probability of

$$\delta_{\texttt{dens}} + \frac{128(\delta_{\texttt{WkL}} + \delta_{\texttt{err}})}{(\eta\gamma^2)} = \delta_{\texttt{dens}} + \delta/6 \leq \delta/3$$

It remains to bound the probability that `Massart-Boost` terminates at round t with $d(\mu_t) > \kappa + \epsilon/2$. Again arguing from the Chernoff-Hoeffding inequality and the sample size of `Est-Density`, if $d(\mu_t) > \kappa + \epsilon/2$, `Massart-Boost` terminates with probability no more than $\delta_{\texttt{dens}}$. Union bounding over all rounds gives a failure probability $128\delta_{\texttt{dens}}/(\eta\gamma^2) = \delta/8$. ■

B.4. Error Bounds

In this subsection we prove upper-bounds for the error of the final hypothesis $H = \text{sign}(G)$, both with respect to the distribution D and to the target function f on the marginal distribution D_x.

Lemma 29 (Label error) *When the algorithm terminates at round t, with all but probability $\delta/4$ over the randomness of* `Massart-Boost`*'s oracles and subroutines,*

$$\text{err}_{0\text{-}1}^{D}(H) \leq \kappa + \epsilon$$

Proof We begin by bounding the error on examples $(x, y) \in \mathcal{X}_t^s$. For all $(x, y) \in \mathcal{X}_t^s$, $H(x) \neq y$ if and only if $yG_t(x) \leq 0$, and therefore $\mu_t(x, y) = 1$. For all other examples, the measure $\mu_t(x, y) \geq 0$. From Lemma 28, we have that with all but probability $\delta/8$, $d(\mu_t) \leq \kappa + \epsilon/2$ upon termination. Conditioning on this event and considering the minimum contribution to the density by all examples misclassified by H, we have

$$\begin{aligned}
\kappa + \epsilon/2 &\geq \mathop{\mathbb{E}}_{(x,y)\sim D} \mu(x,y) \\
&= \sum_{\substack{(x,y):\\ H(x)=y}} D(x,y)\mu(x,y) + \sum_{\substack{(x,y):\\ H(x)\neq y}} D(x,y)\mu(x,y) \\
&\geq \sum_{\substack{(x,y):\\ H(x)\neq y}} D(x,y) \\
&= \mathbf{Pr}_{(x,y)\sim D}[H(x) \neq y \big| x \in \mathcal{X}_t^s].
\end{aligned}$$

We now bound the error of H on examples $(x, y) \in \mathcal{X}_t^r$. When the algorithm terminates at round t, with all but probability $\delta_{\texttt{err}}$, at least one of the following holds.

1. $\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \leq \epsilon/2$
2. $\mathbf{Pr}_{(x,y)\sim D}[H(x)) \neq y \big| x \in \mathcal{X}_t^r] \leq \eta + \epsilon$.

We first consider the case where, in the last round of boosting, `OverConfident` returns false. In this case, either the `OverConfident` routine estimated $\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \leq \epsilon/4$ or it estimated that $\mathbf{Pr}_{(x,y)\sim D}[H(x)) \neq y \big| x \in \mathcal{X}_t^r] \leq \eta + 3\epsilon/4$. The routine uses a sample of size $8\log(2/\delta_{\texttt{err}})/\epsilon^2$ to estimate the probability that $x \in \mathcal{X}_t^r$, and so the probability of underestimating this quantity by more than $\epsilon/4$ is no more than $\delta_{\texttt{err}}/2$, by the Chernoff-Hoeffding inequality. Similarly, the routine uses a sample of size $8\log(2/\delta_{\texttt{err}})/\epsilon^2$ to estimate the error on examples such that $x \in \mathcal{X}_t^r$, and so underestimates this error by more than $\epsilon/4$ with probability no greater than $\delta_{\texttt{err}}/2$. So if `OverConfident` returns false, at least one of the lemma's conditions hold with probability at least $1 - \delta_{\texttt{err}}$.

If `OverConfident` returns true, then $|G_t(x)| = |G_{t-1}(x) + \lambda h_t^s(x)| - \lambda$. From Lemma 25, we know $|G_{t-1}(x)| < s + \lambda$ for all x, and $h_t^s(x) = 0$ for all x such that $|G_{t-1}(x)| \geq s$. It follows that $x \in \mathcal{X}_t^s$ for all x, and so $\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \leq \epsilon/2$.

It remains to bound the total error of H. The error bound for $x \in \mathcal{X}_t^s$ shows

$$\mathrm{err}_{\text{0-1}}^{D}(H) \leq \mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^s](\kappa + \epsilon/2) + \mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \cdot \mathbf{Pr}_{(x,y)\sim D}[H(x) \neq y \big| x \in \mathcal{X}_t^r.]$$

We have also just shown tells us that with all but probability $\delta_{\texttt{err}}$, either

$$\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \leq \epsilon/2$$

or

$$\mathbf{Pr}_{(x,y)\sim D}[H(x) \neq y \big| x \in \mathcal{X}_t^r] \leq \eta + \epsilon.$$

We took $\kappa \geq \eta$, so in either case,

$$\mathrm{err}_{\text{0-1}}^{D}(H) \leq \kappa + \epsilon$$

and so the claimed error with respect to the labels holds except with probability $\delta_{\texttt{err}} + \delta/8 \leq \delta/4$. ■

Lemma 30 (Target function error) *When the algorithm terminates, with all but probability* $\delta/4$*,*

$$\mathrm{err}_{\text{0-1}}^{D_x,f}(H) \leq \frac{\kappa + \epsilon}{1 - \eta}$$

Proof Lemma 29 shows that when the algorithm terminates, with all but probability $\delta/4$,

$$\mathrm{err}_{\text{0-1}}^{D}(H) \leq \kappa + \epsilon,$$

so we consider the worst-case difference between misclassification error and target function error.

$$\begin{aligned}
\kappa + \epsilon &\geq \mathrm{err}_{\text{0-1}}^{D}(H) \\
&= \mathbf{Pr}_{x\sim D_x}[H(x) \neq f(x)] \cdot \mathbf{Pr}_{(x,y)\sim D}[y = f(x) \big| H(x) \neq f(x)] \\
&\qquad + \mathbf{Pr}_{x\sim D_x}[H(x) = f(x)] \cdot \mathbf{Pr}_{(x,y)\sim D}[y \neq f(x) \big| H(x) = f(x)] \\
&\geq \mathbf{Pr}_{x\sim D_x}[H(x) \neq f(x)] \cdot \mathbf{Pr}_{(x,y)\sim D}[y = f(x) \big| H(x) \neq f(x)] \\
&\geq \mathbf{Pr}_{x\sim D_x}[H(x) \neq f(x)](1 - \eta) \\
&= \mathrm{err}_{\text{0-1}}^{D_x,f}(H)(1 - \eta)
\end{aligned}$$

and so $\mathrm{err}_{0\text{-}1}^{D_x,f} \leq \frac{\kappa+\epsilon}{1-\eta}$ with all but probability $\delta/4$. ■

B.5. Sample Complexity Analysis

In this subsection we give sample complexity bounds for the subroutines called by `Massart-Boost`, and the total sample complexity, for a single round of boosting. In all of the following lemmas, we assume that `Massart-Boost` is being run with a (γ, α)-Massart noise weak learner requiring a sample of size $m_{\texttt{WkL}}$. As elsewhere, we use ϵ to denote the target error of the final hypothesis in excess of η, and use κ to denote the density of μ below which `Massart-Boost` terminates. Let $\delta_{\texttt{dens}}$ denote the probability that `Est-Density` fails to estimate the density of μ to within error $\beta = \min\{\epsilon/2, \eta/4\}$ and let $\delta_{\texttt{err}}$ denote the probability that `OverConfident` fails to estimate the error of G_t on examples (x, y) such that $|G_t(x, y)| \geq s$.

Lemma 31 (Sample complexity of `Samp`) *Let failure probability $\delta_{\texttt{samp}} = \delta\eta\gamma^2/(1536\log(2/\delta_{\texttt{WkL}}))$ and $\delta_{\texttt{WkL}} = \delta\eta\gamma^2/1536$. With all but probability $\delta_{\texttt{samp}}$, the `Samp` routine draws no more than*

$$m \in O\left(\frac{\log(1/\delta_{\texttt{Samp}})}{\kappa^2} + \frac{m_{\texttt{WkL}}}{\kappa}\right)$$

examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$.

Proof Because `Massart-Boost` terminates once the density of the measure μ is estimated to be less than κ, and

$$\log(1/\delta_{\texttt{dens}})/(2\beta^2) \geq \max\{2\log(1/\delta_{\texttt{dens}})/\epsilon^2, 8\log(1/\delta_{\texttt{dens}})/\eta^2\}$$

many samples are used to estimate $d(\mu)$, it holds with all but probability $\delta_{\texttt{dens}}$ that $d(\mu) \geq \kappa - \min\{\epsilon/2, \eta/4\}$.

`Samp` terminates once it has kept $m_{\texttt{WkL}}$ examples, and so from Lemma 23 we can conclude that

$$\begin{aligned} m &= \frac{\log(1/\delta_{\texttt{Samp}})}{(\kappa - \min\{\epsilon/2, \eta/4\})^2} + \frac{2m_{\texttt{WkL}}}{\kappa - \min\{\epsilon/2, \eta/4\}} \\ &\in O\left(\frac{\log(1/\delta_{\texttt{Samp}})}{\kappa^2} + \frac{m_{\texttt{WkL}}}{\kappa}\right) \end{aligned}$$

examples suffice except with probability $\delta_{\texttt{Samp}}$. ■

Lemma 32 (Sample complexity of testing weak hypotheses) *Let $\delta_{\texttt{WkL}} = \delta\eta\gamma^2/1536$. With all but probability $\delta\eta\gamma^2/512$, at most*

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right)$$

examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ are drawn to identify a good enough weak hypothesis.

Proof We have just shown in Lemma 31 that, with all but probability $\delta_{\texttt{samp}}$,

$$m \in O\left(\frac{\log(1/\delta_{\texttt{Samp}})}{\kappa^2} + \frac{2m_{\texttt{WkL}}}{\kappa}\right)$$

examples are required to draw a sample for WkL. Recall from Definition 10 that we assume WkL has failure probability 1/3, and from Lemma 20, that we invoke WkL on $2\log(2/\delta_{\texttt{WkL}})$ different samples to ensure we have at least one hypothesis with advantage γ, except with probability $\delta_{\texttt{WkL}}/2$. To estimate which hypothesis is best, we draw $2\log(2/\delta_{\texttt{WkL}})/\gamma^2$ examples from D_μ, against which we test each hypothesis. To draw these additional $2\log(2/\delta_{\texttt{WkL}})/\gamma^2$ examples from D_μ, with all but probability $\delta_{\texttt{samp}}$, we make at most

$$m \in O\left(\frac{\log(1/\delta_{\texttt{samp}})}{\kappa^2} + \frac{\log(1/\delta_{\texttt{WkL}})}{\kappa\gamma^2}\right)$$

calls to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$.

We took $\delta_{\texttt{WkL}} = \delta\eta\gamma^2/1536$ and $\delta_{\texttt{samp}} = \delta\eta\gamma^2/(1536\log(2/\delta_{\texttt{WkL}}))$, so to repeatedly run the weak learner and identify a good enough hypothesis, we require

$$\begin{aligned} m &\in O\left(\frac{\log(1/\delta_{\texttt{WkL}})\log(1/\delta_{\texttt{Samp}})}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/\delta_{\texttt{WkL}})}{\kappa} + \frac{\log(1/\delta_{\texttt{WkL}})}{\kappa\gamma^2}\right) \\ &\in O\left(\frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right) \end{aligned}$$

examples, except with probability

$$\delta_{\texttt{Samp}} + 2\log(2/\delta_{\texttt{WkL}})\delta_{\texttt{samp}} \le 3\log(2/\delta_{\texttt{WkL}})\delta_{\texttt{samp}} = \frac{\delta\eta\gamma^2}{512}.$$

■

Lemma 33 (Sample complexity of `OverConfident`) *With all but probability $\delta\eta\gamma^2/768$, the routine* `OverConfident` *draws no more than*

$$m \in O\left(\frac{\log(1/\delta\eta\gamma)}{\epsilon^3}\right)$$

examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$.

Proof `OverConfident` draws samples to estimate two population statistics: the probability that $x \in \mathcal{X}_t^r$ and, if that estimate exceeds $\epsilon/4$, the error of G_t on examples such that $x \in \mathcal{X}_t^r$.

To estimate $\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r]$ to within error $\epsilon/8$ with all but probability $\delta_{\texttt{err}}/2$, it draws $32\log(2/\delta_{\texttt{err}})/\epsilon^2$ examples. Then to estimate $\mathbb{E}_{(x,y)\sim D}[|y - \mathrm{sign}(G_t(x))| | x \in \mathcal{X}_t^r]$ to within error $\epsilon/4$ with failure probability $\delta_{\texttt{err}}/2$, it uses a sample of size $8\log(2/\delta_{\texttt{err}})/\epsilon^2$, but requires that all these examples satisfy $x \in \mathcal{X}_t^r$. As we know $\mathbf{Pr}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \ge \epsilon/8$ with all but probability

$\delta_{\texttt{err}}/2$, another use of the Chernoff-Hoeffding inequality allows us to upper-bound by $2\delta_{\texttt{err}}$ the probability that `OverConfident` draws more than

$$\begin{aligned} m &= \frac{64\log(1/\delta_{\texttt{err}})}{\epsilon^2} + \frac{128\log(2/\delta_{\texttt{err}})}{\epsilon^3} \\ &\in O\left(\frac{\log(1/\delta_{\texttt{err}})}{\epsilon^3}\right) \\ &\in O\left(\frac{\log(1/\delta\eta\gamma)}{\epsilon^3}\right) \end{aligned}$$

examples to estimate the error.

Therefore with all but probability $2\delta_{\texttt{err}} = \delta\eta\gamma^2/768$, `OverConfident` terminates having drawn no more than

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma)}{\epsilon^3}\right)$$

examples. ■

Lemma 34 (Sample complexity of one round) *With all but probability $5\delta\eta\gamma^2/1536$, one round of boosting with* `WkL` *draws no more than*

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma)}{\min\{\epsilon,\eta\}^2} + \frac{\log(1/(\delta\eta\gamma)}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right)$$

examples from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$.

Proof In a single round of boosting, at most one call is made to `Est-Density` and `OverConfident` routines, and one weak hypothesis is chosen; no calls to the example oracle $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ are otherwise made. The `Est-Density` procedure draws exactly

$$\frac{\log(1/\delta_{\texttt{dens}})}{2\min\{\epsilon/2,\eta/4\}^2} \in O\left(\frac{\log(1/(\delta\eta\gamma)}{\min\{\epsilon,\eta\}^2}\right)$$

examples. Lemma 33 shows that, with all but probability $\delta\eta\gamma^2/768$, the `OverConfident` routine draws no more than

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma)}{\epsilon^3}\right)$$

examples. Lemma 32 shows that, with all but probability $\delta\eta\gamma^2/512$, at most

$$O\left(\frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right)$$

examples are drawn to choose a weak hypothesis. So with all but probability $\delta\eta\gamma^2(\frac{1}{768} + \frac{1}{512}) = 5\delta\eta\gamma^2/1536$, a single round draws no more than

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma)}{\min\{\epsilon,\eta\}^2} + \frac{\log(1/(\delta\eta\gamma)}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right)$$

examples. ■

B.6. Boosting Theorem

We can now put together the Lemmas of Section B.3, Section B.4, and Section B.5 to prove our main result.

Theorem B.1 (Boosting Theorem) *Let* WkL *be an* (α, γ)*-weak learner requiring a sample of size* $m_{\texttt{WkL}}$*. Then for any* $\delta \in (0, 1/2]$*, any Massart distribution* D *with noise rate* $\eta < 1/2$*, and any* $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$*, taking* $\lambda = \gamma/8$ *and* $\kappa = \eta$*,* $\texttt{Massart-Boost}^{\texttt{WkL}}(\lambda, \kappa, \eta, \epsilon, \delta, \gamma, \alpha, m_{\texttt{WkL}})$ *will, with probability* $1 - \delta$*,*

- *run for* $T \in O\left(1/(\eta\gamma^2)\right)$ *rounds*
- *output a hypothesis* H *such that* $\mathrm{err}_{0\text{-}1}^{D}(H) \leq \eta + \epsilon$ *and* $\mathrm{err}_{0\text{-}1}^{D_x,f}(H) \leq \frac{\eta+\epsilon}{1-\eta}$
- *make no more than*

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^3\gamma^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta^2\gamma^2} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2\gamma^4}\right)$$

  calls to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$

- *run in time*

$$O\left(\frac{\log(1/(\delta\eta\gamma))}{\eta^2\gamma^4\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^4\gamma^4} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta^3\gamma^4} + \frac{\log(1/(\delta\eta\gamma))}{\eta^3\gamma^6}\right),$$

  neglecting the runtime of the weak learner.

Proof Lemma 28 shows that `Massart-Boost` terminates within $T \in O\left(1/(\eta\gamma^2)\right)$ rounds, except with probability $\delta/3$. From Lemmas 29 and 30, we have that with all but probability $\delta/4$, $\mathrm{err}_{0\text{-}1}^{D}(H) \leq \kappa + \epsilon$ and $\mathrm{err}_{0\text{-}1}^{D_x,f}(H) \leq \frac{\kappa+\epsilon}{1-\eta}$, so taking $\kappa = \eta$ gives

$$\mathrm{err}_{0\text{-}1}^{D}(H) \leq \eta + \epsilon$$

and

$$\mathrm{err}_{0\text{-}1}^{D_x,f}(H) \leq \frac{\kappa + \epsilon}{1 - \eta}$$

To bound sample complexity, recall Lemma 34 implies that with all but probability $5\delta\eta\gamma^2/1536$, one round of boosting with WkL draws no more than

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma))}{\min\{\epsilon, \eta\}^2} + \frac{\log(1/(\delta\eta\gamma)}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\kappa^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\kappa} + \frac{\log(1/(\delta\eta\gamma))}{\gamma^2\kappa}\right)$$

examples. We have taken $\kappa = \eta$, so union bounding the error probabilities over all $T \leq 128/\eta\gamma^2$ rounds of boosting gives us a sample bound of

$$m \in O\left(\frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^3\gamma^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta^2\gamma^2} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2\gamma^4}\right)$$

exceeded with probability no more than

$$\frac{128}{\eta\gamma^2} \cdot \frac{5\delta\eta\gamma^2}{1536} = \frac{5\delta}{12}.$$

To prove the bound on overall runtime, we first observe that the runtime of a single round of `Massart-Boost`, neglecting calls to the weak learner, is linear in the runtime of subroutines `OverConfident` and `Est-Density`, and quasilinear in the runtime of `Samp` (from repetition of `WkL`). The runtime of each of these subroutines is dominated by computing $G_t(x)$ for each example drawn from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$, either to decide membership of x in $\mathcal{X}_t^r$ or to compute $\mu_t(x, y)$. The cost of evaluating G_t is linear in t, and so from our round and sample complexity bounds, we have the total runtime over all $T \in \left(1/\eta\gamma^2\right)$ rounds `Massart-Boost` is

$$O\left(T\left(\frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^3\gamma^2} + \frac{m_{\mathtt{WkL}}\log(1/(\delta\eta\gamma))}{\eta^2\gamma^2} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2\gamma^4}\right)\right)$$
$$\in O\left(\frac{\log(1/(\delta\eta\gamma))}{\eta^2\gamma^4\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^4\gamma^4} + \frac{m_{\mathtt{WkL}}\log(1/(\delta\eta\gamma))}{\eta^3\gamma^4} + \frac{\log(1/(\delta\eta\gamma))}{\eta^3\gamma^6}\right)$$

Finally, we observe that the total probability of failure to achieve all of the claimed bounds is no more than $\frac{\delta}{3} + \frac{\delta}{4} + \frac{5\delta}{12} = \delta$, completing the proof. ■

B.7. Final Hypothesis H

This subsection contains some explanation of the structure of the final hypothesis H output by our algorithm. We show that these hypotheses can be both efficiently represented and evaluated.

`Massart-Boost` maintains a function $G : \mathcal{X} \to \{\pm 1\}$, initialized to the zero function $G_0(x) = 0$. When `Massart-Boost` terminates at round t, it outputs the classifier $\mathrm{sign}(G_t)$. G_t can be computed from the threshold parameter s and a length-t sequence of pairs $((h_1, b_1), \ldots, (h_t, b_t)) \in (\mathcal{H} \times \{0, 1\})^t$, where h_i is simply the weak learner hypothesis from round i, and $b_i = 1$ if `OverConfident` returned true at round i. $G_t(x)$ can then be efficiently computed by routine 6.

Routine 6 `ComputeG`$(x, s, (h_1, b_1), \ldots, (h_t, b_t))$

```
σ = 0
for i ∈ {1, ..., t} do
    if |σ| < s then
        σ ← σ + λh_i(x)
    else
        if b_i = 1 then
            σ ← σ − λ
return σ
```

Lemma 28 says we may assume $T \in \mathrm{poly}(1/\eta, 1/\gamma)$, so long as the weak learner's hypotheses can be efficiently represented and evaluated, G_T can be as well, and of course $H = \mathrm{sign}(G_T)$.

Appendix C. Improved Round Complexity Analysis

In this section we revisit the round complexity of `Massart-Boost`. We show that a more careful use of the lower-bound on progress against our potential function (Lemma 14) proves convergence in $O\left(\frac{\log^2(1/\eta)}{\gamma^2}\right)$ rounds, saving nearly a factor η^{-1} in both time and sample complexity.

Recall that Lemma 14 shows that in each round of `Massart-Boost` we have

$$\Phi(t) - \Phi(t+1) \geq \frac{\gamma^2}{8}\left(d(\mu_t) - \frac{\eta}{2}\right)$$

for potential function

$$\Phi(t) = \mathop{\mathbb{E}}_{(x,y)\sim D}[\phi_t(x,y)] = \mathop{\mathbb{E}}_{(x,y)\sim D}\int_{yG_t(x)}^{\infty} M(z)dz.$$

For simplicity, Lemma 28 uses the fact that the algorithm terminates once it estimates $d(\mu) \leq \kappa$ to approximately lower-bound $d(\mu_t)$ by κ. Since we take $\kappa \geq \eta$, this lower-bounds the potential drop in each round by $O(\eta\gamma^2)$. However, this lower bound is loose at the beginning of the algorithm, when $d(\mu_0) = 1$. To use this observation to obtain a tighter analysis, we first lower-bound the density of the measure μ_t by the potential function $\Phi(t)$.

Lemma 35 *For every round t, with all but probability $\delta_{\mathtt{err}} = \delta\eta\gamma^2/1536$, $\frac{\Phi_t}{s+\lambda+1} - 2(\eta+\epsilon) \leq d(\mu_t)$.*

Proof To show $\frac{\Phi_t}{s+\lambda+1} - 2(\eta+\epsilon) \leq d(\mu_t)$, we independently consider the contribution to the density from examples $(x,y) \in \mathcal{X}_t^s$ and $(x,y) \in \mathcal{X}_t^r$ as follows,

$$\begin{aligned} d(\mu_t) &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y)] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \big| x \in \mathcal{X}_t^s] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^s] \\ &\qquad + \mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \big| x \in \mathcal{X}_t^r] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^r]. \end{aligned}$$

If $(x,y) \in \mathcal{X}_t^s$, one of two cases holds:

1. $-s < yG_t(x) \leq 0$, so $\mu_t(x,y) = 1$ and $\phi_t(x,y) = -yG_t(x) + 1 \leq s+1$
2. $0 < yG_t(x) < s$, so $\mu_t(x,y) = \exp(-yG_t(x))$ and $\phi_t(x,y) = \exp(-yG_t(x))$

both of which imply $\mu_t(x,y) \geq \phi_t(x,y)/(s+1)$, and so

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \big| x \in \mathcal{X}_t^s] \geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+1} \big| x \in \mathcal{X}_t^s\right] \geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} \big| x \in \mathcal{X}_t^s\right] - 2(\eta+\epsilon).$$

If $(x,y) \in \mathcal{X}_t^r$, then we again have two cases to consider:

1. $yG_t(x) \leq -s$, so $\mu_t(x,y) = 0$ and $\phi_t(x,y) = -yG_t(x) + 1 \leq s+\lambda+1$
2. $yG_t(x) \geq s$, so $\mu_t(x,y) = 0$ and $\phi_t(x,y) = \exp(-yG_t(x)) \leq (\eta+c)/(1-\eta)$.

We observe that examples (x,y) falling into case 2 satisfy

$$\mu_t(x,y) \geq \phi_t(x,y) - (\eta+c)/(1-\eta) \geq \phi_t(x,y) - \frac{\eta+c}{1-\eta},$$

and in case 1, $\mu_t(x,y) \geq \phi_t(x,y)/(s+\lambda+1) - 1$, so to prove our lower-bound on $d(\mu_t)$, we must upper-bound $\mathbf{Pr}_{(x,y)\sim D}[yG_t(x) \leq -s]$. By the definition of Algorithm 2, with all but probability $\delta_{\texttt{err}}$ over the coins of `OverConfident`, at the end of each round t either $\mathbf{Pr}_{x\sim D_x}[x \in \mathcal{X}_t^r] \leq \epsilon/2$ or $\mathbf{Pr}_{(x,y)\sim D}[yG_t(x) \leq -s \mid x \in \mathcal{X}_t^r] \leq \eta + \epsilon$.

If $\mathbf{Pr}_{x\sim D_x}[x \in \mathcal{X}_t^r] \leq \epsilon/2$, then this gives us

$$\begin{aligned} d(\mu_t) &\geq \left(1 - \frac{\epsilon}{2}\right) \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1}\middle| x \in \mathcal{X}^s\right] + \frac{\epsilon}{2} \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} - 1\middle| yG_t(x) \leq -s\right] \\ &\geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1}\right] - \frac{\epsilon}{2} \\ &\geq \frac{\Phi(t)}{s+\lambda+1} - 2(\eta+\epsilon), \end{aligned}$$

and so the stated bound holds.

If $\mathbf{Pr}_{(x,y)\sim D}[yG_t(x) \leq -s \mid x \in \mathcal{X}_t^r] \leq \eta + \epsilon$, we have

$$\begin{aligned} &\mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \middle| x \in \mathcal{X}_t^r] \\ &\quad= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} - 1\middle| yG_t(x) \leq -s\right] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[yG_t(x) \leq -s \middle| x \in \mathcal{X}_t^r] \\ &\qquad\quad + \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\phi_t(x,y) - \frac{\eta+c}{1-\eta}\middle| yG_t(x) \geq s\right] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[yG_t(x) \geq s \middle| x \in \mathcal{X}_t^r] \\ &\quad\geq (\eta+\epsilon) \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} - 1\middle| yG_t(x) \leq -s\right] \\ &\qquad\quad + (1-\eta-\epsilon) \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\phi_t(x,y) - \frac{\eta+c}{1-\eta}\middle| yG_t(x) \geq s\right] \\ &\quad\geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1}\middle| x \in \mathcal{X}_t^r\right] - \eta - \epsilon - (1-\eta-\epsilon)\left(\frac{\eta+c}{1-\eta}\right) \\ &\quad\geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1}\middle| x \in \mathcal{X}_t^r\right] - 2(\eta+\epsilon), \end{aligned}$$

in which case it again holds that

$$
\begin{aligned}
d(\mu_t) &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \big| x \in \mathcal{X}_t^s] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^s] \\
&\quad + \mathop{\mathbb{E}}_{(x,y)\sim D}[\mu_t(x,y) \big| x \in \mathcal{X}_t^r] \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \\
&\geq \left(\mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} \big| x \in \mathcal{X}_t^s \right] - 2(\eta+\epsilon) \right) \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^s] \\
&\quad + \left(\mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{\phi_t(x,y)}{s+\lambda+1} \big| x \in \mathcal{X}_t^r \right] - 2(\eta+\epsilon) \right) \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D}[x \in \mathcal{X}_t^r] \\
&= \frac{\Phi(t)}{s+\lambda+1} - 2(\eta+\epsilon).
\end{aligned}
$$

■

Now that we have a lower-bound on $d(\mu_t)$ in terms of $\Phi(t)$, we can show faster convergence and prove the following theorem.

Theorem C.1 ((Improved) Boosting Theorem) *Let* `WkL` *be an* (α, γ)*-weak learner requiring a sample of size* $m_{\texttt{WkL}}$*. Then for any* $\delta \in (0, 1/2]$*, any Massart distribution* D *with noise rate* $\eta < 1/2$*, and any* $\epsilon \geq \frac{8\eta\alpha}{1-2\alpha}$*, taking* $\lambda = \gamma/8$ *and* $\kappa = \eta$*,* $\texttt{Massart-Boost}^{\texttt{WkL}}(\lambda, \kappa, \eta, \epsilon, \delta, \gamma, \alpha, m_{\texttt{WkL}})$ *will, with probability* $1 - \delta$*,*

- *run for* $T \in O\left(\log^2(1/\eta)/\gamma^2\right)$ *rounds*
- *output a hypothesis* H *such that* $\mathrm{err}_{0\text{-}1}^{D}(H) \leq \eta + \epsilon$ *and* $\mathrm{err}_{0\text{-}1}^{D_x,f}(H) \leq \frac{\eta+\epsilon}{1-\eta}$
- *make no more than* $m \in$

$$O\left(\frac{\log^2(1/\eta)}{\gamma^2}\left(\frac{\log(1/(\delta\eta\gamma))}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta} + \frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2}\right)\right)$$

  calls to $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$*,*
- *run in time*

$$O\left(\frac{\log^4(1/\eta)}{\gamma^4}\left(\frac{\log(1/(\delta\eta\gamma))}{\epsilon^3} + \frac{\log(1/(\delta\eta\gamma))}{\eta^2} + \frac{m_{\texttt{WkL}}\log(1/(\delta\eta\gamma))}{\eta} + \frac{\log(1/(\delta\eta\gamma))}{\eta\gamma^2}\right)\right)$$

  neglecting the runtime of the weak learner.

Proof It follows from Lemma 26, Lemma 14, and Lemma 35 that

$$
\begin{aligned}
\Phi(t+1) &\leq \Phi(t) - \frac{\gamma^2}{32}\left(d(\mu_t) - \frac{\eta}{2}\right) \\
&\leq \Phi(t) - \frac{\gamma^2}{32}\left(\frac{\Phi(t)}{s+\lambda+1} - 2\eta - \frac{\eta}{2}\right) \\
&\leq \Phi(t)\left(1 - \frac{\gamma^2}{64(s+1)}\right) + \frac{\eta\gamma^2}{8} && (\text{ from } s > \lambda).
\end{aligned}
$$

Unrolling the recursion, we have that

$$\Phi(t) \le \left(1 - \frac{\gamma^2}{64(s+1)}\right)^t + \frac{t\eta\gamma^2}{8}$$
$$\le e^{-\gamma^2 t/(64(s+1))} + \frac{t\eta\gamma^2}{8},$$

and so taking $t = 64\log(1/\eta)(s+1)/\gamma^2$ and Lemma 26 gives

$$\begin{aligned} d(\mu_t) &\le \Phi(t) \\ &\le \eta + 8\eta\log(1/\eta)(s+1) \\ &\le \eta + 8\eta\log(1/\eta)(\log(1/\eta)+1) \\ &\le \eta + 24\eta\log^2(1/\eta) \end{aligned}$$

where the last inequality follows from $2\log(1/\eta) > 1$ for all $\eta < 1/2$. As we have already shown a potential drop of at least $\frac{\gamma^2\eta}{32}$ at each step for which $d(\mu) \ge \eta$, running for an additional $768\log^2(1/\eta)/\gamma^2$ rounds suffices to guarantee $d(\mu) \le \eta = \kappa$. This gives a total round complexity of

$$T \in O\left(\frac{\log^2(1/\eta)}{\gamma^2}\right).$$

The stated error bounds are the same as those proved in Theorem B.1, and the tighter sample complexity and runtime follow immediately from the improved round complexity. ■

Appendix D. Lower Bound on Error for Massart Boosting

In this section, we show that no "black-box" generic boosting algorithm for Massart noise can have significantly better error than that of our algorithm, $\eta + \Theta(\alpha\eta)$. While the error term essentially matches the error lower bound of η for RCN boosters from (Kalai and Servedio, 2003), it is unclear from their result whether generalizing to Massart noise should imply a lower bound of OPT or a lower bound of η, since RCN is the special case of Massart noise where $\eta = \mathrm{OPT}$. We show that the lower bound generalizes to the worst-case noise η, so long as OPT is not negligible in the input size. Therefore, no Massart-noise tolerant boosting algorithm can actually take advantage of a distribution with small expected noise to achieve accuracy better than its worst-case noise.

We consider the case where the target function $f \in \mathcal{C}$ is highly biased towards -1 labels (w.l.o.g.) and there is a small fraction of examples $(x, -1)$ where it cannot be distinguished whether $f(x) = 1$ and $\eta(x) = 0$, or $f(x) = -1$ and $\eta(x) > 0$. As described in Section 1.3, if the booster does not reweight the distributions on which it queries the weak learner to emphasize examples labeled 1, an adversarial weak learner can return the constant function -1 and have high correlation. At the same time, if it does reweight its distribution to emphasize positively labeled examples, it risks violating the Massart condition by assigning to some $x \in \mathcal{X}$ a probability of appearing with its noisy label $y = -f(x)$ that is greater than $1/2 - \alpha$.

Our adversarial, "rude" weak learner `rWkL` exploits this tension by providing information attainable without knowledge of f. `rWkL` returns a hypothesis h that does the following: on each heavy-hitter x of given distribution D', $h(x)$ is the majority vote label from examples; on all non-heavy-hitters, $h(x) = -1$. Assuming pseudorandomness of f, no efficient algorithm can use rWkL to learn f. So, the focus of the proof is showing that rWkL is a valid Massart noise weak learner.

By drawing polynomially many more examples than the black-box boosting algorithm, rWkL can *reproducibly* learn the heavy-hitters of its given distribution D'. Furthermore, assuming D' is Massart with noise rate $1/2 - \alpha$, the labels of these heavy hitters $x \in \mathcal{X}$ must be biased towards the true label $f(x)$, guaranteeing advantage on heavy-hitters. To show rWkL also handles non-heavy-hitters correctly, we first show that boosting with rWkL can be efficiently simulated without knowing f, and then we appeal to the pseudorandomness of f. If reweighted distribution D' is Massart, then our adversarial weak learner only fails to return a hypothesis with γ advantage if D' is supported on many non-heavy-hitters x whose true label $f(x) = 1$. We can conclude by observing that finding many such non-heavy-hitters implies a violation of the pseudorandomness assumption.

Theorem 36 *If one-way functions exist, then no black-box Massart noise-tolerant boosting algorithm achieves label error $\eta + o(\alpha\eta)$, even when $\mathrm{OPT} \ll \eta$.*

We formalize the notion of black-box boosting and related definitions in Section D.1. We describe the hard learning problem for the lower bound in Section D.2. We describe our adversarial weak learner and note its useful properties in Section D.3. In Section D.4, we state and prove our lower bound.

D.1. Lower Bound Preliminaries

First, we define black-box boosting. In particular, we formalize the notion of a sampling procedure SP, the subroutine a boosting algorithm uses to construct weak learner queries from labeled examples. Recall the definition of an efficient Massart noise weak learner from Section 2:

Definition 37 (Massart Noise Weak Learner) *Let $\mathcal{C}$ be a concept class of functions $f : \mathcal{X} \to \{\pm 1\}$. Let $\alpha \in [0, 1/2)$. Let $\gamma : \mathbb{R} \to \mathbb{R}$ be a function of α. A Massart noise (α, γ)-weak learner* WkL *for $\mathcal{C}$ is an algorithm such that, for any distribution D_x over $\mathcal{X}$, function $f \in \mathcal{C}$, and noise function $\eta(x)$ with noise bound $\eta < 1/2 - \alpha$,* WkL *outputs a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ such that $\mathbf{Pr}_S[\mathrm{adv}^D(h) \geq \gamma] \geq 2/3$, where the sample S is drawn from Massart distribution $D = \mathtt{Mas}\{f, D_x, \eta(x)\}$.*

Definition 38 (Efficient Massart Noise Weak Learner) *Let* WkL *be an (α, γ)-Massart noise weak learner. Let n be the maximum bit complexity of a single example $(x, y) \in \mathcal{X} \times \{\pm 1\}$, and let $m_{\mathtt{WkL}}$ denote the number of examples comprising sample S.* WkL(S) *is* efficient *if*

1. WkL *uses $m_{\mathtt{WkL}}(n, \eta, \gamma) = \mathrm{poly}(n, 1/(1 - 2\eta), 1/\gamma)$ examples.*
2. WkL *outputs a hypothesis h in time $\mathrm{poly}(n, 1/(1 - 2\eta), 1/\gamma)$.*
3. *Hypothesis $h(x)$ has bit complexity $\mathrm{poly}(n, 1/(1 - 2\eta), 1/\gamma)$.*
4. *For all $x \in \mathcal{X}$, the hypothesis $h(x)$ can be evaluated in time $\mathrm{poly}(n, 1/(1 - 2\eta), 1/\gamma)$.*

We let $CT_{\mathtt{WkL}}$ denote the time WkL takes to output a hypothesis, BC_h denote the maximum bit complexity of a returned hypothesis h, and R_h denote the maximum time to evaluate a returned hypothesis h on any $x \in \mathcal{X}$. Recall that we define the runtime $R_{\texttt{WkL}}$ of WkL as an upper bound on $CT_{\texttt{WkL}} + BC_h + R_h$.

For a boosting algorithm to construct new distributions to query the weak learner, the boosting algorithm must be able to convert examples from D into examples from a new distribution. We refer to this part of the boosting algorithm as a *sampling procedure* SP.

Definition 39 (Sampling Procedure) *A sampling procedure* $\texttt{SP}^{\mathcal{O}}$ *is a probabilistic oracle algorithm that uses (potentially many) examples from* $\mathcal{O}$ *to return an example* $(x, y) \in \mathcal{X} \times \{\pm 1\}$.

We prove a lower bound against the following formulation of a black-box Massart boosting algorithm. In this setting, the boosting algorithm interacts with a example generator EG, which generates examples for the weak learner. The boosting algorithm provides to EG an efficient sampling procedure SP, as well as oracle access to its example oracle EX. The sampling procedure SP will induce a new distribution over $\mathcal{X} \times \{\pm 1\}$. We denote by $D^{\texttt{SP}}$ the distribution induced by SP when supplied with EX as its example oracle. The weak learner WkL uses $m_{\texttt{WkL}}$ examples drawn i.i.d. by EG to compute a hypothesis h, returned to the boosting algorithm. Note that the weak learner is required to return a hypothesis with advantage γ only if $D^{\texttt{SP}}$ is a Massart noise distribution with noise bound $1/2 - \alpha$. For simplicity, we assume that the boosting algorithm and EG know the format of h and SP, and that executing these subroutines can be done efficiently in their respective bit complexities.

Definition 40 (Black-box Massart Boosting Algorithm) *Let* $\mathcal{C}$ *be a concept class over* $\mathcal{X}$*, and let* $f \in \mathcal{C}$ *be an unknown function. Let* n *denote the maximum bit complexity of an* $x \in \mathcal{X}$*. Let* D_x *be a fixed but unknown distribution over* $\mathcal{X}$*. Let* $\mathrm{EX} = \mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ *be a noisy example oracle for Massart noise distribution* $D = \texttt{Mas}\{f, D_x, \eta(x)\}$*. Let* EG *be an example generator with query access to* EX*. Let* WkL *be an efficient* (α, γ)*-Massart noise weak learner with runtime* $R_{\texttt{WkL}}$*, hypothesis bit complexity* BC_h*, and hypothesis evaluation time* R_h*. Let* $m_{\texttt{WkL}}$ *denote the number of examples* WkL *requires. A black-box Massart boosting algorithm* BlackBoxBoost*, with round bound* T *and sample complexity* m*, is a probabilistic polynomial-time algorithm with misclassification error* η^* *if* BlackBoxBoost *satisfies the following conditions:*

1. *Sample complexity* m*:* BlackBoxBoost *draws* $m = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ *examples from sample oracle* EX.

2. *Round bound* T*:* BlackBoxBoost *queries* WkL *at most* $T = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ *times.*

3. *Weak Learner Queries:* BlackBoxBoost *queries* WkL *by providing input* SP *to* EG*, where* $\texttt{SP}^{\mathrm{EX}}$ *is an efficient sampling procedure satisfying the following conditions:*

   - SP *runs in time* $\mathrm{poly}(n, m_{\texttt{WkL}}, 1/(1-2\eta), 1/\gamma, R_h)$.
   - SP *draws at most* $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ *examples from* EX.
   - SP *is represented with bit complexity* $\mathrm{poly}(n, m_{\texttt{WkL}}, 1/(1-2\eta), 1/\gamma, R_h)$.
   - SP *may use previous weak learner hypotheses as subroutines in* SP.

   EG(SP) *runs* SP $m_{\texttt{WkL}}$*-many times to generate a sample* S *containing* $m_{\texttt{WkL}}$ *examples.* EG *gives* S *to* WkL*, which returns a hypothesis* h *to* BlackBoxBoost.

4. *Correctness: If* `WkL` *returns a hypothesis with advantage* γ *over* D^{SP} *in each round that* D^{SP} *is a Massart distribution, then* `BlackBoxBoost` *returns a classifier* $H : \mathcal{X} \to \{\pm 1\}$ *with misclassification error* $\mathrm{err}_{0\text{-}1}^{D}(H) < \eta^*$ *with constant probability.*

5. *Runtime:* `BlackBoxBoost` *runs in time* $\mathrm{poly}(n, m_{\texttt{WkL}}, 1/(1-2\eta), 1/\gamma, R_h)$.

For clarity, the following pseudocode illustrates this black-box boosting framework.

Algorithm 7 Black-box Boosting Framework

Black-box boosting algorithm `BlackBoxBoost` draws m examples from EX.
for $t = 1$ to $t = T$ **do**
  `BlackBoxBoost` constructs sample procedure $\mathtt{SP}_t$, possibly using hypotheses $h_1, \ldots, h_{t-1}$
  `BlackBoxBoost` gives $\mathtt{SP}_t$ to example generator `EG`
  Weak learner `WkL` gives $m_{\texttt{WkL}}$ to `EG`
  `EG` uses $\mathtt{SP}_t$ to draw $m_{\texttt{WkL}}$ i.i.d. examples from $D^{\mathtt{SP}_t}$. Let S denote the set of these examples.
  `EG` gives sample S to `WkL`
  $\mathtt{WkL}(S)$ returns hypothesis h_t to `BlackBoxBoost`
`BlackBoxBoost` outputs trained classifier H

The example generator `EG` is primarily used to correct the type mismatch between the boosting algorithm and weak learner. The boosting algorithm constructs distributions to query the weak learner, and the weak learner is defined to run on samples.

We will show that black-box Massart boosting algorithms cannot learn functions from pseudorandom function families with non-negligible probability. The following definition appears in (Kalai and Servedio, 2003). As noted in (Kalai and Servedio, 2003), if one-way functions exist, then p-biased pseudorandom function families exist.

Definition 41 (p-biased Pseudorandom Function Family) *For* $0 < p < 1$*, a* p*-biased pseudorandom function family is a family of functions* $\{f_s : \{0,1\}^{|s|} \to \{\pm 1\}\}_{s \in \{0,1\}^*}$ *which can be efficiently evaluated and satisfy the following* p*-biased pseudorandomness property:*

- *Efficient evaluation: There is a deterministic algorithm which, given an* n*-bit seed* s *and an* n*-bit input* x*, runs in time* $\mathrm{poly}(n)$ *and outputs* $f_s(x)$*.*

- p*-biased pseudorandomness: Let* $\mathcal{F}_{n,p}$ *be the distribution over functions from* $\{0,1\}^n$ *to* $\{\pm 1\}$ *such that function* F *has weight* $p^{|F^{-1}(1)|}(1-p)^{|F^{-1}(-1)|}$*. For all probabilistic polynomial time algorithms* $\mathcal{A}$*, the distinguishing advantage of* $\mathcal{A}$ *is a negligible function in* n*,*

$$\left| \mathbf{Pr}_{F \sim \mathcal{F}_{n,p}}[\mathcal{A}^F(1^n) \Rightarrow 1] - \mathbf{Pr}_{s \sim \{0,1\}^n}[\mathcal{A}^{f_s}(1^n) \Rightarrow 1] \right| < \mathrm{negl}(n)$$

D.2. Adversarial Massart Distribution

Next, we describe the hard Massart noise learning problem used to prove our lower bound (Theorem 17). The following definitions apply to the remainder of Section 4.

Let $\eta \in [0, 1/2), \alpha \in (0, 1/2-\eta), \gamma(\alpha) = \alpha/20$. Define $\eta' = \eta(1+\alpha/5)$. Let $\{f_s : \{0,1\}^{|s|} \to \{\pm 1\}\}_{s \in \{0,1\}^*}$ be a η'-biased pseudorandom random function family with minority value 1.

Let n denote the security parameter, chosen to be at least a large polynomial in $1/(1-2\eta)$ and $1/\gamma$. Let $\mathcal{X} = \{0,1\}^n$, and let D_x be the uniform distribution over $\mathcal{X}$. For $s \in \{0,1\}^n$, let $\mathcal{C}_s$ be the concept class containing only the function $f_s : \{0,1\}^n \to \{\pm 1\}$.

The noise function $\eta(x)$ is chosen as follows. On the minority elements $x \in f_s^{-1}(1)$, let $\eta(x) = 0$. On the majority elements $x \in f_s^{-1}(-1)$, let $\eta(x) = \eta$ for a random $\rho/(1-\eta')$-fraction of these x's, where $1/\mathrm{poly}(n) < \rho < \alpha/1000$. For the remaining elements, let $\eta(x) = 0$. Finally, let Massart noise distribution $D = \mathtt{Mas}\{f_s, D_x, \eta(x)\}$ with example oracle $\mathrm{EX} = \mathrm{EX}^{\mathrm{Mas}}(f_s, D_x, \eta(x))$. Note that the noise bound is η and $\mathrm{OPT} = \rho\eta$.

Throughout this section, we assume n is a polynomial in $1/(1-2\eta)$ and $1/\gamma$, so that we can assume the probability of EX returning the same data point $x \in \mathcal{X}$ more than once during the $\mathrm{poly}(n, 1/(1-2\eta), \gamma)$ rounds of boosting is a negligible function in n.

D.3. Adversarial Weak Learner and Example Generator

In this section, we describe our adversarial weak learner `rWkL`, provide pseudocode, and prove that it has some nice properties. We also describe an example generator `rEG` that does not directly call EX.

D.3.1. Adversarial Weak Learner

We now define our "rude" weak learner $\mathtt{rWkL}_{m,T}(S)$, which attempts to be maximally unhelpful by returning hypotheses h that rely entirely on majority vote labels. The weak learner `rWkL` never provides the booster with any information about f_s that the booster could not have computed itself, and therefore the pseudorandomness of f_s will guarantee that the booster cannot boost `rWkL` to obtain a hypothesis with error noticeably less than η'. The main technical challenge of proving our lower bound will come from showing that it is in fact possible for `rWkL` to achieve noticeable advantage γ against all Massart distributions supplied to it by the booster, without revealing any information about f_s that cannot be efficiently simulated.

Recall that boosting algorithm `BlackBoxBoost` invokes the weak learner by constructing `SP`, an efficient sampling procedure, which induces a distribution D^{SP}. The weak learner `rWkL` attempts to return a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ satisfying the following two conditions:

- For all $x \in \mathcal{X}$ that have large probability mass in D^{SP} ($\approx \frac{\gamma}{10m}$ or larger), $h(x)$ is the most likely label for x under D^{SP}, i.e., $\mathrm{sign}(\mathbb{E}_{(x^*,y)\sim D^{\mathsf{SP}}}[y \mid x = x^*])$. We will refer to such x's as "heavy-hitters".

- For other x with smaller probability mass in D^{SP}, $h(x)$ is the most likely label for all non-heavy-hitters under D^{SP}, i.e., $\mathrm{sign}(\mathbb{E}_{(x^*,y)\sim D^{\mathsf{SP}}}[y \mid x^* \notin \mathcal{X}^{\mathrm{H}}])$. The weak learner `rWkL` is given access to m, the number of examples drawn by the boosting algorithm, so that `rWkL` may accurately predict which examples x are heavy-hitters.

The weak learner identifies heavy-hitters using a two-step process. First, `rWkL` uses a subset of its sample S to identify candidate heavy-hitters. It initially adds all x-values from this subset to the set of candidate heavy-hitters, $\mathcal{X}^{\mathrm{H}}$. Next, `rWkL` checks each $x \in \mathcal{X}^{\mathrm{H}}$ to see if it is indeed a heavy-hitter of D^{SP}. Fresh examples from its samples S are used to empirically estimate this probability

$\widehat{p}_x \stackrel{\text{def}}{=} \mathbf{Pr}_{(x',y')\sim S^{\mathsf{SP}}}[x = x']$. The weak learner then randomly picks a value $v \in [\frac{\gamma}{20m}, \frac{\gamma}{10m}]$, and removes from $\mathcal{X}^{\mathrm{H}}$ all x's for which $\widehat{p}_x < v$. This step ensures that, with high probability, $\mathcal{X}^{\mathrm{H}}$ contains *exactly* v-heavy-hitters of D^{SP}.

The random choice of v_h and v_y will allow us to argue that, for fixed $v = (v_h, v_y)$, the hypothesis output by `rWkL` is not too sensitive to the specific sample drawn by `rWkL`. That is, if `rWkL` was repeatedly executed with the same choice of v, but different samples drawn from the same distribution, `rWkL` would output the same hypothesis with high probability. This stability property is fully justified in Subsection D.3.4, but, informally, it will allow us to argue that the booster could simulate the example oracle EX itself when generating samples for `rWkL`, without making additional queries to its example oracle, and that with high probability the hypotheses output by `rWkL` would be the same in this case as those output when the sampling procedure queries EX. Analyzing the behavior of the boosting algorithm when the sampling procedure does not draw examples from EX (and therefore the labels of examples do not depend on f_s) simplifies the argument that `rWkL` can satisfy the definition of a Massart noise-tolerant weak learner without leaking information to the booster about f_s.

We now present pseudocode for our adversarial (α, γ)-weak learner.

Algorithm 8 `rWkL`$_{m,T}(S)$

Precondition: S contains $m_{\mathtt{WkL}}$ examples drawn i.i.d. from D^{SP}

if $m < n$ **then**
  $m = n$
// Step 1: Draw candidate heavy-hitters
$\mathcal{X}^{\mathrm{H}} \leftarrow x$-values from $O(m^2/\gamma)$ examples from S
$v_h \leftarrow_r [\frac{\gamma}{20m}, \frac{\gamma}{10m}]$ uniformly at random
for all $x \in \mathcal{X}^{\mathrm{H}}$ **do** // Step 2: Remove non-v-heavy-hitters
  Estimate $\widehat{p}_x \stackrel{\text{def}}{=} \mathbf{Pr}[D^{\mathsf{SP}}$ returns $x]$ using $O(m^{11}T^2/\gamma^4)$ fresh examples from S
  if $\widehat{p}_x < v_h$ **then**
    remove x from $\mathcal{X}^{\mathrm{H}}$
$v_y \leftarrow_r [\frac{1}{2}, \frac{1}{2} + \frac{\gamma}{10m}]$
for all $x \in \mathcal{X}^{\mathrm{H}}$ **do** // Step 3: Assign majority labels
  $S_x \leftarrow m^9T^2/\gamma^4$ fresh examples from S
  $\widehat{p}_1 \leftarrow$ fraction of S_x with label 1
  if $\widehat{p}_1 \geq v_y$ **then**
    $y_x = 1$
  else
    $y_x = -1$
$h(x) = \begin{cases} y_x & x \in \mathcal{X}^{\mathrm{H}} \\ -1 & \text{otherwise} \end{cases}$ // Step 4: Output hypothesis h
return $h \stackrel{\text{def}}{=} \{\mathcal{X}^{\mathrm{H}}, \{y_x\}\}$

This weak learner has polynomial sample complexity (Lemma 42), runs in polynomial time (Lemma 43), and does not use any hardcoded information about f_s, so WkL is efficiently simulatable (Lemma 44).

D.3.2. EXAMPLE GENERATION

In this section, we define the two example generation procedures we will use in our lower bound argument: hEG and rEG.

Recall that an example generator EG is tasked with interfacing between the boosting algorithm, which creates reweighted distributions $D^{\mathtt{SP}}$, and the weak learner, which runs on samples S whose elements are drawn from $D^{\mathtt{SP}}$. To accomplish this, the example generator needs information from the weak learner and the boosting algorithm. The weak learner tells the example generator $m_{\mathtt{WkL}}$, the sample size it needs, and the boosting algorithm provides oracle access to its example oracle EX, as well as the sampling procedure SP. The example generator therefore invokes SP $m_{\mathtt{WkL}}$-many times, returning sample S.

Algorithm 9 $\mathtt{hEG}^{\mathrm{EX}}_{m_{\mathtt{WkL}}}(\mathtt{SP})$
Precondition: SP is a sampling procedure that returns an example (x, y)

```
S = ∅
for i = 1 to i = m_WkL do
    (x, y) ← SP^EX
    S ← S‖(x, y)
return S
```

Our second example generation procedure rEG behaves identically, except it never calls its oracle EX. Rather, rEG simulates calls to EX using EXSim. The routine EXSim draws x values from the same marginal distribution over $\mathcal{X}$ that EX does, $\mathcal{U}(\mathcal{X})$. It then generates the label y by taking $y = -1$ with probability $1 - \eta' - \rho + \rho\eta$, and $y = -1$ otherwise, in effect sampling from the same marginal distribution over ± 1 that EX does, but independent of the value x it has already drawn, and therefore independent of f_s.

Routine 10 EXSim

$x \leftarrow_r U_n$

$y = \begin{cases} -1 & \text{w. p. } 1-\eta'-\rho+\rho\eta \\ 1 & \text{o.w.} \end{cases}$ // i.e. $\mathbf{Pr}[y=1] = \mathbf{Pr}_{(x,y)\sim D}[y=1]$

return (x, y)

Algorithm 11 $\texttt{rEG}^{\text{EX}}_{m_{\texttt{WkL}}}(\texttt{SP})$

Precondition: SP is a sampling procedure that returns an example (x, y)

$S = \emptyset$

for $i = 1$ to $i = m_{\texttt{WkL}}$ **do**

$(x, y) \leftarrow \texttt{SP}^{\text{EXSim}}$

$S \leftarrow S \| (x, y)$

return S

By pseudorandomness, we will show that with high probability over $v = (v_h, v_y)$, and over choice of S, S', where S is generated by hEG and S' is generated by rEG, we have $\texttt{rWkL}(S; v) = \texttt{rWkL}(S'; v)$ (Section D.3.4).

D.3.3. EFFICIENCY OF rWkL AND rEG

In this section, we show that weak learner rWkL and example generator rEG are efficient and simulatable in polynomial time.

1. Efficiency of rWkL: polynomial sample complexity (Lemma 42) and polynomial runtime (Lemma 43).
2. Boosting with rWkL and rEG can be efficienctly simulated (Lemma 44).

Recall that SP is a probabilistic algorithm that returns a labeled example. Let $m_{\texttt{SP}}$ denote the sample complexity of SP. Let $R_{\texttt{SP}}$ denote the runtime of SP (including the time to query its oracle).

Lemma 42 (Sample Complexity of rWkL) $m_{\texttt{rWkL}} = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$.

Proof By Definition 40, $m_{\texttt{SP}} = \text{poly}(n, m, 1/(1-2\eta), 1/\gamma)$, $m = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$, and $T = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$. Step 1 requires $O(m^2T/\gamma)$ examples. Step 2 requires $O(m^{13}T^2/\gamma^5)$ examples. Step 3 requires $O(m^{10}T^2/\gamma^5)$ examples. Therefore Step 2 dominates the sample complexity of the weak learner, and $m_{\texttt{rWkL}} = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$ as claimed. ■

Lemma 43 (Runtime of rWkL) rWkL *runs in time* $R_{\texttt{rWkL}} = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$.

1. rWkL *outputs hypothesis* h *in time* $CT_{\texttt{rWkL}} = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$.
2. *The maximum bit complexity of* h *is* $BC_h = \text{poly}(n, 1/(1-2\eta), 1/\gamma)$.

3. *Hypothesis h can be evaluated in time $R_h = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.*

Proof By Definition 40, $m = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$, and $T = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$. Recall the runtime of a weak learner was defined as a bound on the sum of the three quantities listed in the lemma statement.

The hypotheses $h \stackrel{\text{def}}{=} \{\mathcal{X}^{\mathrm{H}}, \{y_x\}\}$ output by `rWkL` provides individual labels y_x for a maximum of $O(m^2/\gamma)$ elements in $\mathcal{X}^{\mathrm{H}}$. Thus, $\{\mathcal{X}^{\mathrm{H}}, \{y_x\}\}$ has bit complexity at most $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$. The instructions for executing this hypothesis can also be written using $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ bits. For all $x \in \mathcal{X}$, $h(x)$ can be evaluated in time linear in the bit complexity of h. An algorithm can check if $x \in \mathcal{X}^{\mathrm{H}}$ by scanning the representation of h for x, outputting y_x if found or -1 if not. Each step of `rWkL` runs in time linear in the sample complexity of `rWkL`. By Lemma 42, $m_{\texttt{WkL}} = \mathrm{poly}(n, 1/(1-2\eta, 1/\gamma)$. Thus, `rWkL` outputs h in time $\mathrm{poly}(n, 1/(1-2\eta, 1/\gamma)$. ■

Next, we argue that black-box boosting with `rWkL` is efficiently simulatable. The following Lemma permits us to apply use the boosting algorithm in a distinguisher for pseudorandomness.

Lemma 44 (Boosting with `rWkL` and `rEG` can be efficiently simulated) *Given query access to a function oracle for f_s, a probabilistic algorithm $\mathcal{A}$ can simulate `BlackBoxBoost` boosting the weak learner `rWkL`, using `rEG` to generate samples for `rWkL`, in time $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.*

Proof To simulate the initial m examples drawn by the booster, $\mathcal{A}$ simulates EX as follows. It draws a data point $x \in X$ uniformly at random from $\mathcal{X}$, and queries its function oracle on this point. If the label returned by the function oracle is 1, $\mathcal{A}$ returns $(x, 1)$. If the label is a -1, it will return $(x, 1)$ with probability $\rho\eta$, and $(x, -1)$ otherwise. Because $\mathcal{A}$ only has negligible probability of drawing the same x-value twice, and because the noise function $\eta(x)$ is both random and non-zero only on a ρ-sized fraction of negatively-labeled examples, the m examples drawn by this procedure are computationally indistinguishable from m examples drawn from EX, and so $\mathcal{A}$ successfully simulates the initial sample for `BlackBoxBoost`. The algorithm $\mathcal{A}$ can then run the algorithm `BlackBoxBoost`, which by Definition 40, runs in time $R_b = \mathrm{poly}(n, m_{\texttt{rWkL}}, 1/(1-2\eta), 1/\gamma, R_h)$.

To simulate samples generated by `rEG`, $\mathcal{A}$ can simply run Algorithm 11, using Routine 10 for the oracle to the sampling procedure `SP`. We have just shown in Lemma 42 and Lemma 43 that $m_{\texttt{rWkL}}$ and R_h are both $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$, and because the weak learner uses no special hard-coded information about f_s, it can also be efficiently simulated by $\mathcal{A}$. These steps are repeated for $T = \mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting, each of which is efficiently simulatable in time $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$. Any additional post-processing must also be efficiently simulatable, since `BlackBoxBoost` is assumed to run in time $\mathrm{poly}(n, m_{\texttt{WkL}}, 1/(1-2\eta), 1/\gamma, R_h)$, and we have just shown that both $m_{\texttt{WkL}}$ and R_h are $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$. Therefore the entire interaction can be simulated by a probabilistic algorithm $\mathcal{A}$ with a function oracle for f_s, in time $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$.

■

D.3.4. `rWkL` IS A MASSART NOISE-TOLERANT WEAK LEARNER

In this section, we analyze the advantage guarantee of `rWkL`. We begin by proving that the hypotheses output by `rWkL` satisfy some notion of reproducibility. We then use this property, along with pseudorandomness of f_s, to argue that with high probability over choice of sample S generated by hEG, S' generated by rEG, and randomness $v = (v_h, v_y)$, that $\texttt{rWkL}(S; v) = \texttt{rWkL}(S', v)$. We then show that the hypothesis generated by rWkL, when run on a sample generated by rEG, will have good advantage against $D^{\textsf{SP}}$. Therefore the hypothesis generated by rWkL during a real run of the boosting algorithm, where the sample is generated by hEG, must also have good advantage against $D^{\textsf{SP}}$.

Reproducibility of Weak Hypotheses. Recall that rWkL and rEG (or hEG) utilize randomness in two ways: i) to draw the input sample S, and ii) to pick thresholds v_h and v_y. Let h^v be the hypothesis that is most often returned when rWkL is run with thresholds $v = (v_h, v_y)$. In this section, we show that weak learner rWkL has the following property we call *reproducibility*: for a fixed v, with high probability over S, the hypothesis output by $\texttt{rWkL}(S)$ is exactly h^v. We will refer to this hypothesis as the *canonical v-hypothesis* of rWkL.

First, we will show that rWkL run with hEG is reproducible. In the next section, we apply pseudorandomness to show that with high probability, rWkL run with rEG outputs the same canonical v-hypothesis as it does when run with hEG.

Recall that rWkL (Algorithm 8) is designed to return a hypothesis h that assigns majority vote labels to v_h-heavy-hitters of $D^{\textsf{SP}}$, where v_h is randomly chosen in the interval $[\frac{\gamma}{20m}, \frac{\gamma}{10m}]$.

Definition 45 (Heavy-Hitter) *Let D be a distribution over $\mathcal{X} \times \{\pm 1\}$. We call $x \in \mathcal{X}$ a v-*heavy-hitter *of D if $\mathbf{Pr}_{(x^*,y)\sim D}[x = x^*] > v$.*

Recall that rWkL returns a hypothesis $h \stackrel{\text{def}}{=} \{\mathcal{X}^{\text{H}}, \{y_x\}, b\}$. First, we show the consistency of $\mathcal{X}^{\text{H}}$.

Lemma 46 (Consistency of $\mathcal{X}^{\text{H}}$; $\mathcal{X}^{\text{H}}$ is the set of v-heavy-hitters) *Let D be any distribution over $\mathcal{X} \times \{\pm 1\}$, and let sample S be a set of $m_{\texttt{rWkL}}$ examples drawn i.i.d. from D. Then with probability $1 - O(\frac{1}{mT})$ over the choice of S and $v_h \in [\frac{\gamma}{20m}, \frac{\gamma}{10m}]$ (Step 2 of Algorithm 8), the set $\mathcal{X}^{\text{H}}$ computed by $\texttt{rWkL}(S)$ is exactly the set of v_h-heavy-hitters of D.*

Proof Recall that rWkL constructs a candidate list of heavy-hitters $\mathcal{X}^{\text{H}}$ in Step 1 of Algorithm 8, and prunes that list in Step 2.

In Step 1, $\texttt{rWkL}_{m,T}(S)$ uses $O(m^2/\gamma)$ examples to produce the initial set $\mathcal{X}^{\text{H}}$. Let x be a v-heavy-hitter. The probability that $x \notin \mathcal{X}^{\text{H}}$ by the end of Step 1 is at most

$$(1 - \gamma/(20m))^{m^2/\gamma} < \exp(-m/20).$$

Union bounding over the (at most) $20m/\gamma$ v_h-heavy hitters, the probability that the set $\mathcal{X}^{\text{H}}$ does not initially contain all v-heavy hitters is negligible in m.

In Step 2, rWkL estimates $\widehat{p}_x$ for each $x \in \mathcal{X}^{\text{H}}$ using $O(m^{11}T^2/\gamma^5)$ examples. The probability that a sample of this size contains fewer than $O(m^9T^2/\gamma^4)$ instances of x, given that x is a heavy-hitter, is negligible in m, by a Chernoff-Hoeffding bound. Given this many instances of x, the probability that the estimate $\widehat{p}_x$ has error greater than $O(\gamma^2/(m^4T))$ is again a negligible function in m by a Chernoff-Hoeffding bound. Recall rWkL chooses v_h uniformly at random from the interval $[\frac{\gamma}{20m}, \frac{\gamma}{10m}]$. The probability that v_h is chosen to be within distance $O(\gamma^2/(m^4T))$ of the probability of a specific $\gamma/(20m)$-heavy-hitter of $D^{\textsf{SP}}$ is therefore no more than $O(\gamma/(m^3T))$. Union bounding over the at most $20m/\gamma$ heavy hitters, we have the following. Let S_0, S_1 be samples of size $m_{\texttt{rWkL}}$

drawn from D^{SP}. Denote by $\mathcal{X}_0^{\mathrm{H}}(v_h)$ and $\mathcal{X}_1^{\mathrm{H}}(v_h)$ the sets of v_h-heavy-hitters estimated by `rWkL` provided $\mathcal{X}_0^{\mathrm{H}}$ and $\mathcal{X}_1^{\mathrm{H}}$ respectively. Then

$$\Pr_{\substack{S_0,S_1\\ v_h\sim[\frac{\gamma}{20m},\frac{\gamma}{10m}]}}[\mathcal{X}_0^{\mathrm{H}}(v_h)\neq\mathcal{X}_1^{\mathrm{H}}(v_h)]\in O(1/(m^2T)).$$

It remains to show that, with high probability, all non-v-heavy-hitters are not included in $\mathcal{X}^{\mathrm{H}}$ after Step 2. There are at most $O(m^2/\gamma)$ candidate heavy hitters drawn in step 1. With all but negligible probability in m, `rWkL` estimates $\widehat{p}_x$ for all candidate heavy hitters to within error $O(\gamma^2/(m^4T))$. Then, as above, the probability that v_h is chosen to be within distance $O(\gamma^2/(m^4T))$ of the probability of a non-v_h-heavy-hitter of D^{SP} is no more than $O(\gamma/(m^3T))$, and union bounding over the $O(m^2/\gamma)$ candidates gives probability $O(1/(mT))$. Therefore with probability $1-O(\frac{1}{mT})$, at the end of Step 2, $\mathcal{X}^{\mathrm{H}}$ contains exactly the v_h-heavy-hitters of D^{SP}. ■

Next, we show the consistency of $\{y_x\}$, the labels given by `rWkL`(S) to $x\in\mathcal{X}^{\mathrm{H}}$.

Lemma 47 (Reproducibility of h on heavy-hitters) *Let D be a distribution over $\mathcal{X}\times\{\pm1\}$, and let S_0 and S_1 be samples of $m_{\mathtt{rWkL}}$ examples drawn i.i.d. from D. Denote by h_0^v and h_1^v the output of* `rWkL`$(S_0; v_h = v)$ *and* `rWkL`$(S_1; v_h = v)$ *respectively. Let $\mathcal{X}_0^{\mathrm{H}}(v)$ and $\mathcal{X}_1^{\mathrm{H}}(v)$ denote the respective sets of v-heavy-hitters computed by* `rWkL`$(S_0; v_h = v)$ *and* `rWkL`$(S_1; v_h = v)$*. Then we have*

$$\Pr_{\substack{S_0,S_1\\ v\sim[\frac{\gamma}{20m},\frac{\gamma}{10m}]}}[\mathcal{X}_0^{\mathrm{H}}(v)\neq\mathcal{X}_1^{\mathrm{H}}(v)\text{ or }\exists x\in\mathcal{X}_0^{\mathrm{H}}(v)\text{ s.t. }h_0(x)\neq h_1(x)]\in O(1/(mT)).$$

Proof

By Lemma 46, the probability that both $\mathcal{X}_0^{\mathrm{H}}(v)$ and $\mathcal{X}_1^{\mathrm{H}}(v)$ are exactly the set of v-heavy-hitters of D is at least $1-O(1/mT)$, over the choice of v, S_0, and S_1.

For each heavy-hitter x, `rWkL` uses $O(m^{11}T^2/\gamma^4)$ examples to estimate the probability that x has label 1 in D (Step 3 of Algorithm 8). Given that $x\in\mathcal{X}^{\mathrm{H}}$, the probability that this sample contains fewer than $O(m^7T^2/\gamma^4)$ instances of x is negligible in m. By a Chernoff-Hoeffding bound, this estimate has error at most $O(\gamma^2/(m^3T))$ with all but negligible probability in m. By an argument similar to the one of Lemma 46, the probability that v_y falls within $O(\gamma^2/(m^3T))$ of the true probability that x is labeled 1 in D is $O(\gamma/(m^2T))$. Union bounding over the (at most) $20m/\gamma$ heavy-hitters proves the claim

$$\Pr_{\substack{S_0,S_1\\ v\sim[\frac{\gamma}{20m},\frac{\gamma}{10m}]}}[\mathcal{X}_0^{\mathrm{H}}(v)\neq\mathcal{X}_1^{\mathrm{H}}(v)\text{ or }\exists x\in\mathcal{X}_0^{\mathrm{H}}(v)\text{ s.t. }h_0(x)\neq h_1(x)]\in O(1/(mT)).$$

■

Observing that `rWkL` outputs the constant function -1 on all non-heavy-hitters, we have the following corollary.

Corollary 48 (Reproducibility of h) *Let D be a distribution over $\mathcal{X}$, and let samples S_0, S_1 be two sets of $m_{\mathtt{rWkL}}$ examples drawn i.i.d. from D. Let h_0^v and h_1^v denote the hypotheses output by* `rWkL`$(S_0; v)$ *and* `rWkL`$(S_1; v)$ *respectively. Then we have,*

$$\Pr_{S_0,S_1,v}[h_0^v\neq h_1^v]\in O(1/(mT)).$$

We now use the reproducibility of h and the pseudorandomness of $\{f_s\}$ to show that boosting $\texttt{rWkL}$ run with $\texttt{rEG}$ must also output the canonical v-hypothesis with high probability, unless $\mathbf{Pr}_{x\sim D_x^{\texttt{SP}}}[x \notin \mathcal{X}^{\mathrm{H}}] < \gamma$.

Lemma 49 ($\texttt{rWkL}$ does not distinguish between $\texttt{hEG}$ and $\texttt{rEG}$) *Assume $\{f_s\}$ is a pseudorandom function family. Let $\{\texttt{SP}_t\}_{t=1}^{T}$ be a sequence of sampling procedures constructed by the black-box boosting algorithm when boosting $\texttt{rWkL}$ for T rounds. Let $D^{\texttt{SP}_t}$ denote the distribution induced by $\texttt{SP}_t$ and the honest example generator $\texttt{hEG}$, and let $D_r^{\texttt{SP}_t}$ denote the distribution induced by $\texttt{SP}_t$ and the random example generator $\texttt{rEG}$. Let S denote a sample of $m_{\texttt{rWkL}}$ examples drawn i.i.d. from $D^{\texttt{SP}_t}$, and let S_r denote a sample of $m_{\texttt{rWkL}}$ examples drawn i.i.d. from $D_r^{\texttt{SP}_t}$. Let h^v and h_r^v denote the hypotheses output by $\texttt{rWkL}(S;v)$ and $\texttt{rWkL}(S_r;v)$ respectively. Then for all $t \in [T]$,*

$$\mathop{\mathbf{Pr}}_{\substack{S,S_r\\v}}[h_r^v \neq h^v] \in O(1/(mT)).$$

Proof By Corollary 48, we have that $\texttt{rWkL}(S)$ returns the canonical v-hypothesis h^v for $D^{\texttt{SP}_t}$ with high probability over choice of v and S. Then if the claim does not hold, then it must be the case that there exists a round $t \in [T]$ such that, with probability $\omega(1/(mT))$ over choice of v, S, and S_r, we have $\texttt{rWkL}(S;v) \neq \texttt{rWkL}(S_r;v)$. Assuming this, we can construct the following distinguisher $\mathcal{A}$ against the pseudorandomness of $\{f_s\}$.

The distinguisher $\mathcal{A}$ executes the following procedure. It first chooses a round $t \in [T]$ uniformly at random, and simulates the interaction between the booster and $\texttt{rWkL}$ until round t. At round t, $\mathcal{A}$ draws a sample S_0 of $m_{\texttt{rWkL}}$ examples from $D^{\texttt{SP}_t}$ by simulating $\texttt{hEG}$. It then draws a sample S_1 of $m_{\texttt{rWkL}}$ examples by simulating $\texttt{rEG}$. It simulates $\texttt{rWkL}$ on both of these samples using the same choice of randomness v for both simulations, and checks whether $\texttt{rWkL}(S_0;v) = \texttt{rWkL}(S_1;v)$. If not, it returns 1, and otherwise returns 0.

In the case that $\mathcal{A}$ is give oracle access to a random function F, both S_0 and S_1 are drawn from the same distribution, and so by Corollary 48, $\texttt{rWkL}(S_0;v) = \texttt{rWkL}(S_1;v)$ with probability $1 - O(1/(mT))$ over the choice of v, and therefore $\mathcal{A}$ outputs 1 with probability $O(1/(mT))$.

In the case that $\mathcal{A}$ is supplied a pseudorandom function f_s, by assumption there exists a round $t \in [T]$ at which $\mathbf{Pr}_{\substack{S_0,S_1\\v}}[\texttt{rWkL}(S_0;v) \neq \texttt{rWkL}(S_1;v)] \in \omega(1/(mT))$. Therefore in this case, $\mathcal{A}$ outputs 1 with probability noticeably (in n) larger than in the random case, and so $\mathcal{A}$ is a distinguisher against the pseudorandomness of f_s. This is a contradiction, and therefore the claim holds. ■

Informally, Lemma 49 will allow us to construct distinguishing adversaries against the pseudorandomness of f_s that make only m queries of their function oracle. In the following lemmas, we will prove that $\texttt{rWkL}$ satisfies the definition of a Massart noise-tolerant weak learner when invoked on distributions constructed by the booster. That is, when $\texttt{rWkL}$ is given a sample from a Massart distribution generated by the boosting algorithm, it returns a weak hypothesis with advantage γ with probability at least $2/3$. We will rely on appeals to the pseudorandomness of f_s in these proofs, by showing that failure of $\texttt{rWkL}$ to return hypothesis with good advantage allows for the construction of distinguishers against the pseudorandomness of f_s. These distinguishers will simulate the boosting procedure, but it will be useful for our proofs to claim that the distinguishers can generate samples for $\texttt{rWkL}$ without making additional queries to their function oracles to generate labels for these samples. Lemma 49 allows us to design distinguishers that use $\texttt{rEG}$ to generate samples for $\texttt{rWkL}$, rather than generating samples using hEG. Recall that rEG makes no calls to the example oracle EX, and simply generates labels randomly for examples drawn from the underlying marginal distribution D_x. Therefore we will assume that our distinguishers only query their function oracles for the purposes of simulating the first m examples drawn by the booster.

Advantage of rWkL. We will prove the following lemma by separately considering the advantage of weak hypotheses on heavy hitters of D^{SP} and non-heavy hitters.

Lemma 50 (Advantage of rWkL) *Let D^{SP_t} denote the distribution induced by the sampling procedure SP_t and hEG at round $t \in [T]$ of boosting. Similarly, let $D_r^{\mathsf{SP}_t}$ denote the distribution induced by SP_t and rEG. Let S_t denote a sample drawn i.i.d. from $D_r^{\mathsf{SP}_t}$. Then for all $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting rWkL with rEG, if D^{SP_t} is Massart, then with probability $1 - O(1/(mT))$ over its internal randomness, $\mathtt{rWkL}(S_t)$ outputs a hypothesis h_t with advantage at least γ against D^{SP_t}, except with negligible probability in m over the choice of SP_t.*

Recall that n is chosen to be a polynomial in $1/(1-2\eta)$ and $1/\gamma$, and D_x is the uniform distribution over $\{0,1\}^n$. By birthday-paradox-style arguments, with all but negligible probability in n, no $x \in \mathcal{X}$ is output more than once by EX throughout boosting. Henceforth, we assume no $x \in \mathcal{X}$ is output more than once by EX.

Lemma 51 (rWkL advantage against heavy-hitters of D^{SP}) *Let D^{SP_t} be the distribution induced by the sampling procedure SP_t at round t. Similarly, let $D_r^{\mathsf{SP}_t}$ denote the distribution induced by SP_t and rEG. Let S_t denote a sample drawn i.i.d. from $D_r^{\mathsf{SP}_t}$, and let h_t be the hypothesis output by $\mathtt{rWkL}(S_t)$. Then for all $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting rWkL with rEG, either*

1. $\mathbf{Pr}\left[\frac{1}{2}\,\mathbb{E}_{(x,y)\sim D^{\mathsf{SP}_t}}[h_t(x)y \mid x \in \mathcal{X}^{\mathrm{H}}] \geq \alpha\right] \in 1 - O(1/(mT))$
2. *or D^{SP_t} is not Massart.*

Proof From reproducibility of h_t (Lemma 49), we have that with probability $1 - O(1/(mT))$, rWkL outputs the same hypothesis that it would have had it been given a sample from D^{SP_t}. For the remainder of the proof then, we will analyze the behavior of rWkL given such a sample from D^{SP_t}, and show that it must have good advantage against the heavy-hitters of D^{SP_t}.

Suppose the second case does not hold, and therefore D^{SP_t} is Massart. To compute y_x for each $x \in \mathcal{X}^{\mathrm{H}}$, rWkL uses m^9T^2/γ^4 examples from $D_r^{\mathsf{SP}_t}$. Because $x \in \mathcal{X}^{\mathrm{H}}$, $D_x^{\mathsf{SP}_t}(x) \geq \gamma/(40m)$ with high probability, and taking $\gamma = \alpha/20$, we have that at least $4m/\alpha^2$ instances of x occur in S_x (Step 3 of Algorithm 8) with all but negligible probability in m. The majority label of these $4m/\alpha^2$ examples is then taken to be the prediction of h_t on x, which will agree with $f(x)$ with all but negligible probability in m, because we have assumed D^{SP_t} is Massart, and so

$$\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[y = f(x) \mid x \in \mathcal{X}^{\mathrm{H}}] \geq 1/2 + \alpha.$$

It then follows that

$$\mathbf{Pr}\left[\frac{1}{2}\mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[h_t(x)y \mid x \in \mathcal{X}^{\mathrm{H}}] < \alpha\right] \leq \mathrm{negl}(m)$$

or D^{SP_t} is not Massart, when h_t is the hypothesis output by `rWkL` given a sample S from D^{SP_t}. Applying Lemma 49 allows us to conclude that

$$\mathbf{Pr}\left[\frac{1}{2}\underset{(x,y)\sim D^{\mathsf{SP}_t}}{\mathbb{E}}[h_t(x)y \mid x \in \mathcal{X}^{\mathrm{H}}] < \alpha\right] \leq O(1/(mT))$$

or D^{SP_t} is not Massart, when $h_t \leftarrow \mathtt{rWkL}(S_t)$. ■

Lemma 52 (`rWkL` advantage against non-heavy hitters of D^{SP}) *Let D^{SP_t} be the distribution induced by the sampling procedure SP_t at round t. Similarly, let $D_r^{\mathsf{SP}_t}$ denote the distribution induced by SP_t and `rEG`. Let S_t denote a sample drawn i.i.d. from $D_r^{\mathsf{SP}_t}$, and let h_t be the hypothesis output by $\mathtt{rWkL}(S_t)$. Then for all $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting `rWkL` with `rEG`, with all but negligible probability in m over choice of SP_t, either*

1. $\mathbf{Pr}_{S_t,v}\left[\frac{1}{2}\mathbb{E}_{(x,y)\sim D^{\mathsf{SP}_t}}[h_t(x)y \mid x \notin \mathcal{X}^{\mathrm{H}}] < \gamma\right] < 1/\mathrm{poly}(n)$
2. $\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}] < \gamma$
3. *or D^{SP_t} is not Massart*

Proof

Suppose that the first two conditions fail, implying that there exists a round t of boosting for which the advantage of h_t on the non-heavy hitters of D^{SP_t} is less than γ, and that this will noticeably impact the overall advantage. Because h_t takes a constant value -1 on all non-heavy hitters, it must then be the case that

$$\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[y = 1 \mid x \notin \mathcal{X}^{\mathrm{H}}] > 1/2 - \gamma.$$

Since we are considering the advantage only on examples such that $x \notin \mathcal{X}^{\mathrm{H}}$, then $D^{\mathsf{SP}_t}(x, y) < \gamma/(10m)$ for all these examples. Furthermore, since we have assumed $\sum_{(x,y):x\notin\mathcal{X}^{\mathrm{H}}} D^{\mathsf{SP}_t}(x, y) \geq \gamma$, there must be at least $5m$ non-heavy-hitter examples x such that $D^{\mathsf{SP}_t}(x, 1) > D^{\mathsf{SP}_t}(x, -1)$ in order for D^{SP_t} to satisfy $\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[y = 1 \mid x \notin \mathcal{X}^{\mathrm{H}}] > 1/2 - \gamma$. Then for D^{SP_t} to be Massart, it must hold that $f(x) = 1$ for every example x such that $D^{\mathsf{SP}_t}(x, 1) > D^{\mathsf{SP}_t}(x, -1)$. However, if this is true with non-negligible probability in n, then we can construct the following distinguisher against f_s, which we denote by $\mathcal{A}$.

The distinguisher $\mathcal{A}$ simulates the boosting procedure run with `rWkL` and `rEG`, as described in Lemma 44 up until round t, chosen uniformly at random from $[1, T]$. Once the boosting procedure reaches round t, $\mathcal{A}$ simulates the t'th round of boosting and then queries its function oracle on all examples from the sample of the weak learner at that round that satisfy $D^{\mathsf{SP}_t}(x, 1) > D_x^{\mathsf{SP}_t}(x)(1/2 - \alpha)$. If $f(x) = 1$ for all these examples, $\mathcal{A}$ outputs 1, and otherwise outputs 0.

To lower bound the advantage of our distinguisher, we will first show that there must be a significant number of examples drawn by `rWkL` in round t that satisfy $D^{\mathsf{SP}_t}(x, 1) > D_x^{\mathsf{SP}_t}(x)(1/2 - \alpha)$. We begin by lower bounding the probability that this condition holds for a single non-heavy hitter example.

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[D^{\mathsf{SP}_t}(x,1) > D_x^{\mathsf{SP}_t}(x)(1/2-\alpha) \mid x \notin \mathcal{X}^{\mathrm{H}}] \\
&\qquad = 1 - \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[D^{\mathsf{SP}_t}(x,1) \leq D_x^{\mathsf{SP}_t}(x)(1/2-\alpha) \mid x \notin \mathcal{X}^{\mathrm{H}}] \\
&\qquad = 1 - \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[D^{\mathsf{SP}_t}(x,-1) \geq D_x^{\mathsf{SP}_t}(x)(1/2+\alpha) \mid x \notin \mathcal{X}^{\mathrm{H}}] \\
&\qquad \geq 1 - \frac{1+2\gamma}{1+2\alpha} \\
&\qquad = \frac{2(\alpha-\gamma)}{1+2\alpha} \\
&\qquad \geq \alpha - \gamma \\
&\qquad > \alpha/2,
\end{aligned}$$

where the last line follows from taking $\gamma = \alpha/20$, and the third line follows from observing that

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[D^{\mathsf{SP}_t}(x,-1) \geq D_x^{\mathsf{SP}_t}(x)(1/2+\alpha) \mid x \notin \mathcal{X}^{\mathrm{H}}] &= \sum_{\substack{x\notin\mathcal{X}^{\mathrm{H}} \\ D^{\mathsf{SP}_t}(x,-1)\geq D_x^{\mathsf{SP}_t}(x)(1/2+\alpha)}} D^{\mathsf{SP}_t}(x) \cdot \frac{1}{\mathbf{Pr}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}]} \\
&\leq \sum_{\substack{x\notin\mathcal{X}^{\mathrm{H}} \\ D^{\mathsf{SP}_t}(x,-1)\geq D_x^{\mathsf{SP}_t}(x)(1/2+\alpha)}} \frac{D^{\mathsf{SP}_t}(x,-1)}{1/2+\alpha} \cdot \frac{1}{\mathbf{Pr}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}]} \\
&\leq \frac{\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[y=-1 \mid x \notin \mathcal{X}^{\mathrm{H}}]}{1/2+\alpha}.
\end{aligned}$$

We have assumed that $\mathbf{Pr}_{(x,y)\sim D^{\mathsf{SP}_t}}[y=-1 \mid x \notin \mathcal{X}^{\mathrm{H}}] < 1/2+\gamma$, and so

$$\mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[D^{\mathsf{SP}_t}(x,-1) \geq D_x^{\mathsf{SP}_t}(x)(1/2+\alpha) \mid x \notin \mathcal{X}^{\mathrm{H}}] \leq \frac{1+2\gamma}{1+2\alpha}.$$

Then because `rWkL` has a sample of size $O(m^{13}T^2/\gamma^5)$, we have that the probability that fewer than n of them satisfy $D^{\mathsf{SP}_t}(x,1) > D_x^{\mathsf{SP}_t}(x)(1/2-\alpha)$ must be negligible in m by the Chernoff-Hoeffding inequality.

We now proceed to bound the distinguishing advantage of $\mathcal{A}$, beginning with $\mathbf{Pr}_{F\sim\mathcal{F}_{n,\eta'}}[\mathcal{A}^F \Rightarrow 1]$. In the case that f is a random function, the boosting procedure can correctly identify an x such that $f(x)=1$ with probability no greater than $\frac{\eta'}{\eta'+(1-\eta')\rho\eta}$. This follows immediately from taking the largest of the following conditional probabilities:

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{(x,y)\sim D^{\mathsf{SP}_t}}[f(x)=1 \mid y=1] = \frac{\eta'}{\eta'+(1-\eta')\rho\eta} \\
&\mathop{\mathbf{Pr}}_{(x,y)\sim D^{\mathsf{SP}_t}}[f(x)=1 \mid y=-1] = 0 \\
&\mathop{\mathbf{Pr}}_{(x,y)\sim D^{\mathsf{SP}_t}}[f(x)=1 \mid y \text{ unknown }] = \eta'.
\end{aligned}$$

So if f is a truly random function then the boosting procedure has probability no more than $(\frac{\eta'}{\eta'+(1-\eta')\rho\eta})^n$ of correctly identifying at least n non-heavy hitter preimages of 1 under f. We have just shown that with all but negligible probability, `rWkL` draws at least n examples satisfying $D^{\mathtt{SP}_t}(x,1) > D_x^{\mathtt{SP}_t}(x)(1/2-\alpha)$, and $\mathcal{A}$ returns 1 only if all of these examples are preimages of 1 under f. Therefore

$$\Pr_{F\sim\mathcal{F}_{n,\eta'}}[\mathcal{A}^F \Rightarrow 1] \leq \left(\frac{\eta'}{\eta'+(1-\eta')\rho\eta}\right)^n + \mathrm{negl}(n),$$

where the additive $\mathrm{negl}(n)$ term comes from the probability that fewer than n qualifying examples were drawn by `rWkL` in that round.

We now consider the case that $\mathcal{A}$ is provided f_s as its oracle. Towards contradiction we have assumed that there exists some round t at which, with probability p that is non-negligible in m, the booster produces a Massart distribution $D^{\mathtt{SP}_t}$ for which $\mathbf{Pr}_{(x,y)\sim D^{\mathtt{SP}_t}}[y=1 \mid x \notin \mathcal{X}^{\mathrm{H}}] > 1/2-\gamma$. Therefore with probability $1/T$ the distinguisher $\mathcal{A}$ will halt its simulation at this round, and so with probability p/T will produce such a distribution. Then with all but negligible probability, it will draw n examples such that $D^{\mathtt{SP}_t}(x,1) \geq D_x^{\mathtt{SP}_t}(x)(1/2-\alpha)$. Since the distribution is Massart, all of these examples must satisfy $f(x)=1$, and so we have

$$\Pr_{s\sim\{0,1\}^n}[\mathcal{A}^{f_s} \Rightarrow 1] = p/T - \mathrm{negl}(m),$$

which is non-negligible in m, and therefore n. Therefore $\mathcal{A}$ has distinguishing advantage

$$\Pr_{s\sim\{0,1\}^n}[\mathcal{A}^{f_s} \Rightarrow 1] - \Pr_{F\sim\mathcal{F}_{n,\eta'}}[\mathcal{A}^F \Rightarrow 1] > \mathrm{negl}(n),$$

which contradicts pseudorandomness of $\mathcal{F}_{n,\eta'}$. Therefore it must be the case that the boosting procedure only has negligible probability (in m) of generating a Massart distribution at any round that has at least γ probability mass assigned to non-heavy hitters, and for which the constant function -1 does not have advantage at least γ against non-heavy-hitters of $D^{\mathtt{SP}_t}$.

■

We can now combine Lemma 51 and Lemma 52 to show that `rWkL`, given a sample generated by `rEG`, will output a hypothesis with good advantage against $D^{\mathtt{SP}_t}$.

Lemma 53 (Advantage of `rWkL`) *Let $D^{\mathtt{SP}_t}$ denote the distribution induced by the sampling procedure $\mathtt{SP}_t$ and `hEG` at round $t \in [T]$ of boosting. Similarly, let $D_r^{\mathtt{SP}_t}$ denote the distribution induced by $\mathtt{SP}_t$ and `rEG`. Let S_t denote a sample drawn i.i.d. from $D_r^{\mathtt{SP}_t}$. Then for all $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ rounds of boosting `rWkL` with `rEG`, if $D^{\mathtt{SP}_t}$ is Massart, then with probability $1-O(1/(mT))$ over its internal randomness, $\mathtt{rWkL}(S_t)$ outputs a hypothesis h_t with advantage at least γ against $D^{\mathtt{SP}_t}$, except with negligible probability in m over the choice of $\mathtt{SP}_t$.*

Proof

The advantage of h_t against $D^{\mathtt{SP}_t}$ is $\frac{1}{2}\,\mathbb{E}_{(x,y)\sim D^{\mathtt{SP}_t}}[yh(x)]$ where

$$
\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x)] &= \mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x) \mid x \in \mathcal{X}^{\mathrm{H}}] \cdot \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[x \in \mathcal{X}^{\mathrm{H}}] \\
&\quad + \mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x) \mid x \notin \mathcal{X}^{\mathrm{H}}] \cdot \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}] \\
&\geq \alpha \cdot (1 - \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}]) + \mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x) \mid x \notin \mathcal{X}^{\mathrm{H}}] \cdot \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}] \\
&= \alpha - \mathop{\mathbf{Pr}}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}] \cdot (\alpha - \mathop{\mathbb{E}}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x) \mid x \notin \mathcal{X}^{\mathrm{H}}]),
\end{aligned}
$$

with all but probability $O(1/(mT))$, following from Lemma 51. From Lemma 52, we have that if D^{SP_t} is Massart, then with all but negligible probability, either $\mathbb{E}_{(x,y)\sim D^{\mathsf{SP}_t}}[yh(x) \mid x \notin \mathcal{X}^{\mathrm{H}}] \geq \gamma$ or $\mathbf{Pr}_{x\sim D_x^{\mathsf{SP}_t}}[x \notin \mathcal{X}^{\mathrm{H}}] < \gamma$. Therefore h has advantage at least γ against D^{SP_t} with probability at least $1 - O(1/(mT))$, and the claim holds. ■

D.4. Lower Bound for Black-Box Massart Boosting

Finally, we prove that no black-box boosting algorithm can boost `rWkL` to misclassification error better than $\eta(1 + o(\alpha))$ with noticeable probability. At a high level, the proof idea is that any black-box booster interacting with `rWkL` can be efficiently simulated, and so if a boosting algorithm was able to achieve misclassification error noticeably better than $\eta(1 + o(\alpha))$ for $\{f_s\}$, then there must be a distinguisher against the pseudorandomness of this function family, and so such error cannot be achievable via black-box boosting algorithms so long as pseudorandom functions exist.

Theorem 17 (Error Lower Bound Theorem) *Let $\eta \in [0, 1/2), \alpha \in (0, 1/2 - \eta)$. Let $\{f_s\}$ be an η'-biased pseudorandom function family with security parameter n, where $\eta' = \eta(1 + \alpha/5)$. Let η, α be at least inversely polynomially in n bounded away from $1/2$. Then, for random s, no efficient black-box boosting algorithm* `BlackBoxBoost` *with example bound m running for T rounds, given query access to $(\alpha, \gamma(\alpha) \stackrel{\text{def}}{=} \alpha/20)$-weak learner $\mathtt{rWkL}_{m,T}$ and $\mathrm{poly}(n, 1/(1-2\eta), 1/\gamma)$ examples from example oracle $\mathrm{EX}(U_n, f_s, \eta(x))$, can output a hypothesis with label error at most $\eta(1 + o(\alpha))$. In particular, for all polynomials q, for all polynomial time black-box Massart boosting algorithms* `BlackBoxBoost` *with query access to* `rWkL` *and example oracle* EX*, for n sufficiently large, $\mathbf{Pr}_{s\in U_n}\left[\mathrm{err}_{0\text{-}1}^{U_n,f_s}(H) \leq \eta'\right] < \frac{1}{q(n)}$, where H is the trained classifier output by* `BlackBoxBoost`.

Proof [Proof of Theorem 17]

Let $\eta' = \eta(1 + c\alpha)$. Suppose that `BlackBoxBoost` achieves label error better than $\eta' - \epsilon$, for some noticeable ϵ, and with noticeable probability δ. Then we can construct a distinguisher $\mathcal{A}$ for f_s as follows.

The distinguisher $\mathcal{A}$ simulates the interaction between the booster and `rWkL`, where the samples for `rWkL` are drawn by `rEG` (as described in Lemma 44). Once the booster outputs its final hypothesis H, $\mathcal{A}$ draws a set S of n/ϵ^2 elements from the uniform distribution over $\mathcal{X}$, restricted to examples on which it has not already queried its oracle. Because `rWkL` is being run on samples drawn by `rEG`, $\mathcal{A}$ will only have simulated EX, and therefore queried its oracle, for the m examples used by the booster itself, and therefore n/ϵ^2 elements can be drawn efficiently and the restricted distribution has only negligible statistical distance from D_x. The distinguisher $\mathcal{A}$ then queries both H and its oracle on all elements of S, returning 1 if its oracle and H disagree on fewer than an $\eta' - \epsilon/2$ fraction of the elements, and 0 otherwise.

To show that $\mathcal{A}$ has non-negligible advantage distinguishing f_s from a truly random function, we first consider the probability that $\mathcal{A}$ outputs 1 when given oracle access to a truly random function, drawn from $\mathcal{F}_{n,\eta'}$. Because $\mathcal{A}$ is checking $H(x) \neq f(x)$ only on examples it has not previously queried, once H is fixed, we have $\mathbf{Pr}_{x\sim\mathcal{U}(\mathcal{X})}[H(x) \neq f(x) \mid x \text{ not previously queried}] \geq \eta'$. Therefore

$$\begin{aligned}\mathop{\mathbf{Pr}}_{F\sim\mathcal{F}_{n,\eta'}}[\mathcal{A}^F \Rightarrow 1] &= \mathop{\mathbf{Pr}}_{\substack{F\sim\mathcal{F}_{n,\eta'}\\ S\sim\mathcal{U}(\mathcal{X})}}[\mathop{\mathbf{Pr}}_{x\sim S}[H(x) \neq F(x)] \leq \eta' - \epsilon/2] \\ &\leq \text{negl}(n),\end{aligned}$$

where the last line follows from a Chernoff-Hoeffding bound and the fact that $\mathcal{A}$ has drawn n/ϵ^2 elements from $\mathcal{X}$ to check.

We now consider the probability that $\mathcal{A}$ returns 1 when given oracle access to pseudorandom f_s. We have assumed that our booster has noticeable probability δ of outputting a hypothesis H with error less than $\eta' - \epsilon$, and from Lemma 49, we have that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{s\sim\{0,1\}^n}[\mathcal{A}^{f_s} \Rightarrow 1] &= \mathop{\mathbf{Pr}}_{\substack{s\sim\{0,1\}^n\\ S\sim\mathcal{U}(\mathcal{X})}}[\mathop{\mathbf{Pr}}_{x\sim S}[H(x) \neq f_s(x)] \leq \eta' - \epsilon/2] \\ &\geq \delta(1 - \text{negl}(n)).\end{aligned}$$

Since we have assumed δ is noticeable, and we have just shown that $\mathcal{A}$ has distinguishing advantage

$$\mathop{\mathbf{Pr}}_{s\sim\{0,1\}^n}[\mathcal{A}^{f_s} \Rightarrow 1] - \mathop{\mathbf{Pr}}_{F\sim\mathcal{F}_{n,\eta'}}[\mathcal{A}^F \Rightarrow 1] > \delta/2,$$

the distinguisher $\mathcal{A}$ contradicts the pseudorandomness of f_s, and therefore `rWkL` cannot be efficiently boosted to construct a hypothesis with error noticeably better than η' with any noticeable probability.

■

Appendix E. Application: Massart Learning of Unions of High-Dimensional Rectangles

In this section, we exhibit a Massart weak learner for learning unions of rectangles and apply our boosting algorithm.

Definition 54 (Rectangle) *A rectangle $B \in \mathbb{R}^d$ is an intersection of inequalities of the form $x \cdot v < t$, where $v \in \{\pm e_j : j \in [d]\}$ and $t \in \mathbb{R}$. We may write a rectangle as a set B of pairs (v, t), that has size at most $2d$.*

We are interested in learning concepts $f \in C$ that are indicator functions of unions of k rectangles $B_1, \ldots, B_k$. That is, the class $\mathcal{C}$ consists of functions:

$$f(x) = \begin{cases} +1 & \text{if } x \in \cup_{i\in[k]} \cap_{(v,t)\in B_i} [x \cdot v < t] \\ -1 & \text{otherwise} \end{cases}$$

We refer to the negation of $\cup_{i\in[k]} B_i$ as the "negative region". Our weak learner aims to find if possible a rectangle entirely contained in the negative region to get some advantage over a random guess. To this end, we establish a structural result which shows that unless an overwhelming part of the mass is positive, there always exists a rectangle with non-trivial mass that is contained in the negative region. Moreover this rectangle has a lot of structure as it consists of at most k inequalities.

Lemma 55 (Structural result) *If the negative region has probability more than ε, there exists a rectangle contained in the negative region that has mass at least $\varepsilon/(2d)^k$. This rectangle can be written as an intersection of at most k inequalities.*

Proof The negative region can be written as a union of $(2d)^k$ rectangles B' with at most k inequalities

$$\cup_{B' \in B_1 \times B_2 \times \cdots \times B_k} \cap_{(v,t)\in B'} [x.v \geq t]$$

by choosing which inequality is not satisfied in every rectangle.

Since the union of the rectangles covers is exactly the negative region and has mass at least ε, at least one rectangle B' has probability more than $\varepsilon/(2d)^k$. ■

E.1. Weak Learner

Our weak learner exploits the structural result of Lemma 55 to obtain an advantage over a random guess. If the probability mass is overwhelmingly positive, then the hypothesis $h(x) = +1$ must correlate well with the observed labels. On the contrary, if there is sufficient negative mass, there must exist a rectangle where predicting $h(x) = -1$ correlates with the labels of the examples within that rectangle. This idea is presented in pseudo-code in $\texttt{WkL}_{\texttt{box}}$ and formalized in Lemma 56 which gives the guarantees of our weak learner.

Algorithm 12 $\texttt{WkL}_{\texttt{box}}{}^{\mathrm{EX}(f,D_x,\eta(x))}(d,k,\alpha)$

$S \leftarrow \frac{k\,O(d)^k}{\alpha^2}$ examples from EX
$S^- \leftarrow$ number of these examples labeled -1
if $\frac{|S^-|}{|S|} < \frac{\alpha}{2}$ **then**
  return $h = +1$ // the constant 1 hypothesis
else
  for all Rectangles B = choice of k examples and k dimensions **do**
    $S_B \leftarrow \{(x,y) \in S | x \in B\}$
    $S_B^+ \leftarrow \{(x,y) \in S | x \in B, y = +1\}$
    $B_{best} \leftarrow B$ that minimizes $|S_B^+|/|S_B|$ and has $|S_B|/|S| > \frac{\alpha}{8(2d)^k}$.
  Let $z \in \{\pm 1\}$ be the best most popular label in $S \setminus S_{B_{best}}$
  Hypothesis $h(x) = \begin{cases} -1 & x \in B_{best} \\ z & \text{otherwise} \end{cases}$
  return h

Lemma 56 *The algorithm* $\texttt{WkL}_{\texttt{box}}$ *is a* $(\alpha, \frac{\alpha^2}{O(d)^k})$*-Weak Learner for unions of* k *rectangles in* d *dimensions. It requires* $k\frac{O(d)^k}{\alpha^2}$ *samples and runs in time* $\frac{k^k O(d)^{k^2+1}}{\alpha^{2k}}$.

Proof The algorithm starts by drawing drawing a set S of $N = k\frac{O(d)^k}{\alpha^2}$ examples from EX. Since the VC-dimension of rectangles defined by k inequalities is $O(k)$ this guarantees that, with probability at least $2/3$, for any rectangle B, the empirical probabilities computed over the sample S are close to actual ones:

1. $|\mathbf{Pr}[x \in B] - \mathbf{Pr}_S[x \in B]| \leq \alpha/O(d)^k$
2. $|\mathbf{Pr}[y = +1 \text{ and } x \in B'] - \mathbf{Pr}_S[y = +1 \text{ and } x \in B']| \leq \alpha/O(d)^k$
3. $|\mathbf{Pr}[y = -1] - \mathbf{Pr}_S[y = -1]| \leq \alpha/O(d)^k \leq \frac{\alpha}{4}$

Therefore, in the case that $|S^-|/|S| < \frac{\alpha}{2}$, we have that $\mathbf{Pr}[y = -1] < \frac{3}{4}\alpha$. Thus, the hypothesis $h = +1$ gets error at most $\frac{3}{4}\alpha + (\frac{1}{2} - \alpha) \leq \frac{1}{2} - \frac{\alpha}{4}$.

Otherwise, there is at least $\frac{\alpha}{4}$ probability in the negative region. By Lemma 55, there is a rectangle B^* defined by k inequalities that is contained entirely in the negative region and has probability at least $\frac{\alpha}{4(2d)^k}$. For this rectangle B^* it holds that $\mathbf{Pr}[x \in B^*] \geq \frac{\alpha}{4(2d)^k}$ and $\mathbf{Pr}[y = +1 | x \in B] \leq \frac{1}{2} - \alpha$. This means that within the sample S it holds that $\mathbf{Pr}_S[x \in B^*] > \frac{\alpha}{8(2d)^k}$ and $\mathbf{Pr}_S[y = +1 | x \in B^*] \leq \frac{1}{2} - \frac{\alpha}{2}$. Thus, B_{best} will also satisfy $\mathbf{Pr}_S[y = +1 | x \in B_{best}] \leq \frac{1}{2} - \frac{\alpha}{2}$. By the closeness guarantee of the empirical distribution, we get that $\mathbf{Pr}[y = +1 | x \in B_{best}] \leq \frac{1}{2} - \frac{\alpha}{4}$ and $\mathbf{Pr}[x \in B_{best}] > \frac{\alpha}{9(2d)^k}$.

We now bound the error of the hypothesis

$$h(x) = \begin{cases} -1 & x \in B_{best} \\ z & \text{otherwise} \end{cases}$$

Within the region B_{best}, it achieves error at most $\frac{1}{2} - \frac{\alpha}{4}$, while outside of B_{best}, the error is at most. $\frac{1}{2} + \frac{\alpha}{O(d)^k}$. Thus, the total error is at most $\frac{1}{2} - \frac{\alpha^2}{O(d)^k}$ given that $\mathbf{Pr}[x \in B_{best}] > \frac{\alpha}{9(2d)^k}$.

The main computational step of the algorithm is searching over all rectangles with k inequalities. It suffices to only consider rectangles with samples as end points, thus the total runtime of the weak-learner is $O(dN)^k = \frac{k^k O(d)^{k^2+1}}{\alpha^{2k}}$ as for every inequality there are $2d$ choices for the direction v and N choices for the threshold t.

■

E.2. Putting Everything Together

Lemma 56 shows that $\mathtt{WkL_{box}}$ is an $(\alpha, \frac{\alpha^2}{O(d)^k})$-Weak Learner for unions of k high-dimensional rectangles in d dimensions. Combined with Theorem 16 we get that:

Theorem 57 *There exists an algorithm that learns unions of k rectangles in d dimensions with Massart noise bounded by η, achieving misclassification error $\eta + \epsilon$ for $\epsilon > 0$. The total number of samples is $\frac{kd^{O(k)}}{\eta^2 \epsilon^8}$ and the total running time is $\frac{1}{\eta^3}\left(\frac{kd^k}{\epsilon}\right)^{k+O(1)}$.*

Proof Follows by a direct application of the weak learner to Theorem 16 for $\alpha = \epsilon/8$ and $\gamma = \frac{\epsilon^2}{O(d)^k}$.
■

Appendix F. Glossary of Symbols

Problem Statement

$\mathcal{X}$ A large finite domain

D_x A distribution over $\mathcal{X}$

$\mathcal{C}$ A class of concepts from $\mathcal{X}$ to $\{\pm 1\}$

f The unknown function in $\mathcal{C}$ to be learned

$\eta(x)$ The Massart noise function

η The Massart noise parameter, an upper bound on the Massart noise function

$D = \mathtt{Mas}\{f, D_x, \eta(x)\}$ A Massart distribution over $\mathcal{X} \times \{\pm 1\}$

$\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ The noisy example oracle

Weak Learners

$\mathtt{WkL}$ The (α, γ)-weak learner to be boosted

h A hypothesis returned by the weak learner

α The Massart noise tolerance of the weak learner $(1/2 - \alpha)$

γ The advantage of the weak learner

S A sample, i.e., a collection of labeled examples, $S \in (X \times \{\pm 1\})^m$

$m_{\texttt{WkL}}$ The sample complexity of the weak learner

$\delta_{\texttt{WkL}}$ The failure rate of the weak learner

Boosting Algorithm

$\eta + \epsilon$ The target error rate of the boosting algorithm (PAC-learning parameter)

δ The target failure rate of the boosting algorithm (PAC-learning parameter)

$G : \mathcal{X} \to \mathbb{R}$ Determines the final classifier. Updated in each round of boosting

λ The learning rate of the boosting algorithm, chosen $\Theta(\gamma)$.

T The number of rounds of boosting (and weak learner queries)

$t \in [T]$ A single round of boosting, commonly used as a subscript

`Samp`, `Est-Density`, `OverConfident` The subroutines of boosting algorithm `Massart-Boost`

Boosting Algorithm – Reweighting Distributions

$\mu : X \times \{\pm 1\} \to [0, 1]$ A "measure" function used to determine rejection sampling probabilities

D_μ The distribution induced by rejection sampling from $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$ according to μ

$d(\mu)$ The density of μ, $d(\mu) = \mathbb{E}_{(x,y)\sim D}[\mu(x, y)]$

κ The density below which the algorithm terminates. Needs to be larger than η for our potential argument. We want $\kappa \approx \eta$, since the algorithm cannot get error better than $\kappa + \epsilon$.

$\mathcal{X}^r$ The set of "risky" $x \in \mathcal{X}$, i.e. $\{x \in \mathcal{X} \mid |G(x)| \geq s\}$. The weak learner is never given examples from $\mathcal{X}^r$ (i.e. $\mu(x, y) = 0$ if $x \in \mathcal{X}^r$), to ensure that the induced distribution D_μ is a Massart distribution with noise rate $(1/2 - \alpha)$.

$\mathcal{X}^s$ The set of "safe" $x \in \mathcal{X}$, i.e. $\{x \in \mathcal{X} \mid |G(x)| < s\}$.

$s = \log\left(\frac{1-\eta}{\eta+c}\right)$ The cutoff between risky and safe regions of $G(x)$

$c = \frac{4\eta\alpha}{1-2\alpha}$ A constant used to limit the noise rate of reweighted distributions

Boosting Algorithm – Analysis

Φ The global potential function $\mathbb{E}_{(x,y)\sim D}[\phi(x, y)]$

$\phi(x, y)$ The potential function of an example (x, y)

$M : \mathcal{X} \to [0, 1]$ The "base" measure function, used to define both $\phi(x, y)$ and $\mu(x, y)$

Lower Bound

`rWkL` The adversarial, unboostable, "rude" weak learner

`BlackBoxBoost` A black-box boosting algorithm. For each weak learner query, `BlackBoxBoost` generates a sampling procedure `SP` and passes it to the example generator

`SP` A sampling procedure, i.e. an efficient routine to generate a sample S using $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$

$D^{\mathtt{SP}}$ The distribution induced by `SP` and $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$

`EG` An example generator. Generates a sample $S \sim_{\text{i.i.d.}} D^{\mathtt{SP}}$ using `SP`, and passes S to the weak learner. Resolves a type mismatch between boosting algorithms (which generate a distribution to query the weak learner) and weak learners (which take as input a sample drawn from a distribution)

`hEG` An "honest" example generator. Runs `SP` without any alterations.

`rEG` A "rude" example generator. Runs `SP`, but simulates sampling from $\mathtt{Mas}\{f, D_x, \eta(x)\}$ without using $\mathrm{EX}^{\mathrm{Mas}}(f, D_x, \eta(x))$. Used in conjunction with `rWkL` in the lower bound construction

η' The error lower bound parameter, $\eta' = \eta(1 + \Theta(\alpha))$

ρ The fraction of examples that are noisy in the lower bound construction.

OPT The average noise rate $\mathbb{E}_{(x,y)\sim D}[\eta(x)]$. In the lower bound construction, $\mathrm{OPT} = \rho\eta$

$\mathcal{X}^{\mathrm{H}}$ The set of heavy-hitters of distribution D

h^v The canonical v-hypothesis of `rWkL`.