Efficient, Noise-Tolerant, and Private Learning via Boosting

Mark Bun MBUN@BU.EDU
Department of Computer Science
Boston University
111 Cummington Mall, Boston MA 02215, USA

Marco Leandro Carmosino MARCO@NTIME.ORG
School of Computing Science
Simon Fraser University
8888 University Drive, Burnaby, BC V5A 1S6, Canada

Jessica Sorrell JLSORREL@UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA

Editors: Jacob Abernethy and Shivani Agarwal

Abstract

We introduce a simple framework for designing private boosting algorithms. We give natural conditions under which these algorithms are differentially private, efficient, and noise-tolerant PAC learners. To demonstrate our framework, we use it to construct noise-tolerant and private PAC learners for large-margin halfspaces whose sample complexity does not depend on the dimension.

We give two sample complexity bounds for our large-margin halfspace learner. One bound is based only on differential privacy, and uses this guarantee as an asset for ensuring generalization. This first bound illustrates a general methodology for obtaining PAC learners from privacy, which may be of independent interest. The second bound uses standard techniques from the theory of large-margin classification (the fat-shattering dimension) to match the best known sample complexity for differentially private learning of large-margin halfspaces, while additionally tolerating random label noise.

Keywords: Boosting, Differential Privacy, Learning with Noise, Linear Threshold Functions

1. Introduction

1.1. (Smooth) Boosting, Noise-Tolerance, and Differential Privacy

Boosting is a fundamental technique in both the theory and practice of machine learning for converting weak learning algorithms into strong ones. Given a sample S of n labeled examples drawn i.i.d. from an unknown distribution, a weak learner is guaranteed to produce a hypothesis that can predict the labels of fresh examples with a noticeable advantage over random guessing. The goal of a boosting algorithm is to convert this weak learner into a strong learner: one which produces a hypothesis with classification error close to zero.

A typical boosting algorithm — e.g., the AdaBoost algorithm of Freund and Schapire (1997) — operates as follows. In each of rounds $t = 1, \ldots, T$, the boosting algorithm selects a distribution D_t over S and runs the weak learner on S weighted by D_t, producing a hypothesis h_t. The history of hypotheses $h_1, \dots, h_t$ is used to select the next distribution D_{t+1} according to some update rule (e.g., the multiplicative weights update rule in AdaBoost). The algorithm terminates either after a fixed number of rounds T, or when a weighted majority of the hypotheses $h_1, \dots, h_T$ is determined to have sufficiently low error.

In many situations, it is desirable for the distributions D_t to be *smooth* in the sense that they do not assign too much weight to any given example, and hence do not deviate too significantly from the uniform distribution. This property is crucial in applications of boosting to noise-tolerant learning (Domingo and Watanabe, 2000; Servedio, 2003), differentially private learning (Dwork et al., 2010), and constructions of hard-core sets in complexity theory (Impagliazzo, 1995; Barak et al., 2009b). Toward the first of these applications, Servedio (2003) designed a smooth boosting algorithm (`SmoothBoost`) suitable for PAC learning in spite of malicious noise. In this model of learning, up to an η fraction of the sample S could be corrupted in an adversarial fashion before being presented to the learner (Valiant, 1985). Smooth boosting enables a weak noise-tolerant learner to be converted into a strong noise-tolerant learner. Intuitively, the smoothness property is necessary to prevent the weight placed on corrupted examples in S from exceeding the noise-tolerance of the weak learner. The round complexity of smooth boosting was improved by Barak et al. (2009b) to match that of the AdaBoost algorithm by combining the multiplicative weights update rule with Bregman projections onto the space of smooth distributions.

Smoothness is also essential in the design of boosting algorithms which guarantee *differential privacy* (Dwork et al., 2006), a mathematical definition of privacy for statistical data analysis. Kasiviswanathan et al. (2011) began the systematic study of PAC learning with differential privacy. Informally, a (randomized) learning algorithm is differentially private if the distribution on hypotheses it produces does not depend too much on any one of its input samples. Again, it is natural to design "private boosting" algorithms which transform differentially private weak learners into differentially private strong learners. In this context, smoothness is important for ensuring that each weighted input sample does not have too much of an effect on the outcomes of any of the runs of the weak learner. A private smooth boosting algorithm was constructed by Dwork et al. (2010), who augmented the AdaBoost algorithm with a private weight-capping scheme which can be viewed as a Bregman projection.

1.2. Our Contributions

Simple and Modular Private Boosting. Our main result is a framework for private boosting which simplifies and generalizes the private boosting algorithm of Dwork et al. (2010). Our framework is flexible enough to accommodate adaptations of Servedio's `SmoothBoost` algorithm (Appendix K), as well as the smooth boosting algorithm of Barak et al. (2009a) based on Bregman projections (Section 4). We obtain these simplifications by sidestepping a technical issue confronted by Dwork et al. (2010). Their algorithm maintains two elements of state from round to round: the history of hypotheses $H = h_1, \dots, h_t$ and auxiliary information regarding each previous distribution $D_1, \dots, D_t$, which is used to enforce smoothness. They remark: "[this algorithm] raises the possibility that adapting an existing or future smooth boosting algorithm to preserve privacy might yield a simpler algorithm."

We realize exactly this possibility by observing that most smooth boosting algorithms have effectively *stateless* strategies for re-weighting examples at each round. By definition, a boosting algorithm must maintain some history of hypotheses. Therefore, re-weighting strategies that can be computed using only the list of hypotheses require no auxiliary information. Happily, most smooth boosting algorithms define such hypothesis-only re-weighting strategies. Eliminating auxiliary state greatly simplifies our analysis, implies natural conditions under which existing smooth boosting algorithms can be easily privatized, and yields lower sample complexity. A detailed comparison between our boosting algorithm and that of Dwork et al. (2010) appears in Appendix L.

Our main algorithm is derived from that of Barak et al. (2009a), which we call `BregBoost`. Their algorithm alternates between mutiplicative re-weighting and Bregman projection: the multiplicative update reflects current performance of the learner, and the Bregman projection ensures that `BregBoost` is smooth. Unfortunately, a naïve translation of `BregBoost` into our framework would Bregman project more than once per round. This maintains correctness, but ruins privacy. Inspired by the private optimization algorithms of Hsu et al. (2013, 2014) we give an alternative analysis of `BregBoost` that requires only a *single* Bregman projection at each round. The need for "lazy" Bregman projections emerges naturally by applying our template for private boosting to `BregBoost`, and results in a private boosting algorithm with optimal round complexity: `LazyBregBoost`. This method of lazy projection (see Rakhlin, 2009, for an exposition) has appeared in prior works about differential privacy (Hsu et al., 2013, 2014), but not in the context of designing boosting algorithms.

Application: Privately Learning Large-Margin Halfspaces. A halfspace is a function $f : \mathbb{R}^d \to \{-1, 1\}$ of the form $f(x) = \text{sign}(u \cdot x)$ for some vector $u \in \mathbb{R}^d$. Given a distribution D over the unit ball in $\mathbb{R}^d$, the *margin* of f with respect to D is the infimum of $|u \cdot x|$ over all x in the support of D. Learning large-margin halfspaces is one of the central problems of learning theory. A classic solution is given by the Perceptron algorithm, which is able to learn a τ-margin halfspace to classification error α using sample complexity $O(1/\tau^2\alpha)$ independent of the dimension d. Despite the basic nature of this problem, it was only in very recent work of Nguyễn et al. (2019) that dimension-independent sample complexity bounds were given for *privately* learning large-margin halfspaces. In that work, they designed a learning algorithm achieving sample complexity $\tilde{O}(1/\tau^2\alpha\varepsilon)$ for τ-margin halfspaces with $(\varepsilon, 0)$-differential privacy, and a computationally efficient learner with this sample complexity for (ε, δ)-differential privacy. Both of their algorithms use dimensionality reduction (i.e., the Johnson-Lindenstrauss lemma) to reduce the dimension of the data from d to $O(1/\tau^2)$. One can then learn a halfspace by privately minimizing the hinge-loss on this lower dimensional space.

Meanwhile, one of the first applications of smooth boosting was to the study of noise-tolerant learning of halfspaces. Servedio (2003) showed that smooth boosting can be used to design a (non-private) algorithm with sample complexity $\tilde{O}(1/\tau^2\alpha^2)$ which, moreover, tolerates an $\eta = O(\tau\alpha)$ rate of malicious noise. Given the close connection between smooth boosting and differential privacy, it is natural to ask whether private boosting can also be used to design a learner for large-margin halfspaces. Note that while one could pair the private boosting algorithm of Dwork, Rothblum, and Vadhan with our differentially private weak learner for this application, the resulting hypothesis would be a majority of halfspaces, rather than a single halfspace. Like Nguyễn et al. (2019) we address *proper* differentially-private learning of large-margin halfspaces where the hypothesis is itself a halfspace and not some more complex Boolean device.

We use our framework for private boosting to achieve a proper halfspace learner with sample complexity $\tilde{O}\left(\frac{1}{\varepsilon\alpha\tau^2}\right)$ when (ε, δ)-DP is required. Our learner is simple, efficient, and automatically tolerates random classification noise (Angluin and Laird, 1987) at a rate of $O(\alpha\tau)$. That is, we recover the sample complexity of Nguyễn et al. (2019) using a different algorithmic approach while also tolerating noise.[1] Additionally, our efficient algorithm guarantees *zero-concentrated* differential privacy (Bun and Steinke, 2016), a stronger notion than (ε, δ)-DP. In this short paper we phrase all guarantees as (ε, δ)-DP to facilitate comparison of sample bounds.

Theorem 1 (Informal, Fat-Shattering Application to Large-Margin Halfspaces) *Given $n = \tilde{O}\left(\frac{1}{\varepsilon\alpha\tau^2}\right)$ samples from a distribution D supported by a τ-margin halfspace u subject to $O(\alpha\tau)$-rate random label noise, our learning algorithm is (ε, δ)-DP and outputs with probability $(1-\beta)$ a halfspace that α-approximates u over D.*

Furthermore, it may be interesting that we can also obtain non-trivial sample bounds for the same problem using *only* differential privacy. The analysis of Nguyễn et al. (2019) uses the VC dimension of halfspaces and the analyses of Servedio (2003) and Theorem 1 above both use the fat-shattering dimension of halfspaces to ensure generalization. We can instead use the generalization properties of differential privacy to prove the following (in Appendix H).

Theorem 2 (Informal, Privacy-Only Application to Large-Margin Halfspaces) *Given $n = \tilde{O}\left(\frac{1}{\varepsilon\alpha\tau^2} + \varepsilon^{-2} + \alpha^{-2}\right)$ samples from a distribution D supported by a τ-margin halfspace u subject to $O(\alpha\tau)$-rate random label noise, our learning algorithm is (ε, δ)-DP and outputs with probability $(1-\beta)$ a halfspace that α-approximates u over D.*

Intuitively, the fat-shattering argument has additional "information" about the hypothesis class and so can prove better bounds. However, the argument based only on differential privacy would apply to *any* hypothesis class with a differentially private weak learner. So, we present a template for generalization of boosting in Sections G.2 and G.3 which relies *only* on the learner's privacy guarantees.

2. Preliminaries

2.1. Measures & Distributions

For a finite set X, let $\mathcal{U}(X)$ be the uniform distribution over X.

Definition 3 (Bounded Measures) *A* bounded measure *on domain X is a function $\mu : X \to [0,1]$.*

- **Density:** $d(\mu) = \mathbb{E}_{x\sim\mathcal{U}(X)}\left[\mu(x)\right]$ — *the "relative size" of a measure in X.*
- **Absolute Size:** $|\mu| = \sum_{x\in X} \mu(x)$
- **Induced Distribution:** $\hat{\mu}(x) = \mu(x)/|\mu|$ — *the distribution obtained by normalizing a measure*

We require some notions of similarity between measures and distributions.

Definition 4 (Kullback-Leibler Divergence) *Let μ_1 and μ_2 be bounded measures over the same domain X. The* Kullback-Leibler divergence *between μ_1 and μ_2 is defined as:*

$$\mathrm{KL}(\mu_1 \parallel \mu_2) = \sum_{x\in X} \mu_1(x) \log\left(\frac{\mu_1(x)}{\mu_2(x)}\right) + \mu_2(x) - \mu_1(x)$$

[1] Our algorithm only tolerates "local" noise patterns like RCN, because malicious noise could ruin privacy. See Section G.3 of the appendix for a discussion.

Definition 5 (Statistical Distance) *The* statistical distance *between two distributions Y and Z, denoted $\Delta(Y, Z)$, is defined as:*

$$\Delta(Y, Z) = \max_S |\Pr[Y \in S] - \Pr[Z \in S]|$$

The α-Rényi divergence has a parameter $\alpha \in (1, \infty)$ which allows it to interpolate between KL-divergence at $\alpha = 1$ and max-divergence at $\alpha = \infty$.

Definition 6 (Rényi Divergence) *Let P and Q be probability distributions on Ω. For $\alpha \in (1, \infty)$, we define the Rényi Divergence of order α between P and Q as:*

$$D_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} \log \left(\mathop{\mathbb{E}}_{x \sim Q} \left[\left(\frac{P(x)}{Q(x)} \right)^\alpha \right] \right)$$

A measure is "nice" if it is simple and efficient to sample from the associated distribution. If a measure has high enough density, then rejection sampling will be efficient. So, the set of *high density* measures is important, and will be denoted by:

$$\Gamma_\kappa = \{\mu \mid d(\mu) \geq \kappa\}.$$

To maintain the invariant that we only call weak learners on measures of high density, we use Bregman projections onto the space of high density measures.

Definition 7 (Bregman Projection) *Let $\Gamma \subseteq \mathbb{R}^{|S|}$ be a non-empty closed convex set of measures over S. The* Bregman projection *of $\tilde{\mu}$ onto Γ is defined as:*

$$\Pi_\Gamma \tilde{\mu} = \arg\min_{\mu \in \Gamma} \mathrm{KL}(\mu \parallel \tilde{\mu})$$

Bregman projections have the following desirable property:

Theorem 8 (Bregman, 1967) *Let $\tilde{\mu}, \mu$ be measures such that $\mu \in \Gamma$. Then,*

$$\mathrm{KL}(\mu \parallel \Pi_\Gamma \tilde{\mu}) + \mathrm{KL}(\Pi_\Gamma \tilde{\mu} \parallel \tilde{\mu}) \leq \mathrm{KL}(\mu \parallel \tilde{\mu}). \textit{ In particular, } \mathrm{KL}(\mu \parallel \Pi_\Gamma \tilde{\mu}) \leq \mathrm{KL}(\mu \parallel \tilde{\mu}).$$

Barak, Hardt, and Kale gave the following characterization of Bregman projections onto the set of κ-dense measures, which we will also find useful.

Lemma 9 (Bregman Projection onto Γ_κ is Capped Scaling Barak et al. (2009a)) *Let Γ denote the set of κ-dense measures. Let $\tilde{\mu}$ be a measure such that $|\tilde{\mu}| < \kappa n$, and let $c \geq 1$ be the smallest constant such that the measure μ, where $\mu(i) = \min\{1, c \cdot \tilde{\mu}(i)\}$, has density κ. Then $\Pi_\Gamma \tilde{\mu} = \mu$.*

2.2. Learning

We work in the **P**robably **A**pproximately **C**orrect (**PAC**) setting (Valiant, 1984). Two types of error are allowed: first, the algorithm is allowed to completely fail with some small probability, and second, it outputs a hypothesis that is only *close* to the concept we are trying to learn. We denote by $\mathcal{X}$ the domain of examples, and for the remainder of this work consider only the Boolean classification setting where labels are always ±1.

Definition 10 (PAC Learning) *A hypothesis class $\mathcal{H}$ is (α, β)-*PAC learnable *if there exists a sample bound $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ such that: for every $\alpha, \beta \in (0,1)$ and for every distribution D over $\mathcal{X} \times \{\pm 1\}$, running $\mathcal{A}$ on $n \geq n_{\mathcal{H}}(\alpha, \beta)$ i.i.d. samples from D will with probability at least $(1-\beta)$ return a hypothesis $h : \mathcal{X} \to \{\pm 1\}$ such that:*

$$\Pr_{(x,y)\sim D}[h(x) = y] \geq 1 - \alpha.$$

PAC learners guarantee strong generalization to unseen examples. We will construct PAC learners by boosting weak learners — which need only beat random guessing on any distribution over the training set.

Definition 11 (Weak Learning) *Let $S \subset (\mathcal{X} \times \{\pm 1\})^n$ be a training set of size n. Let D be a distribution over $[n]$. A* weak learning algorithm *with* advantage *γ takes (S, D) as input and outputs a function $h : \mathcal{X} \to [-1, 1]$ such that:*

$$\frac{1}{2}\sum_{j=1}^{n} D(j)|h(x_j) - y_j| \leq \frac{1}{2} - \gamma$$

2.3. Privacy

Two datasets $S, S' \in X^n$ are said to be *neighboring* (denoted $S \sim S'$) if they differ by at most a single element. Differential privacy requires that analyses performed on neighboring datasets have "similar" outcomes. Intuitively, the presence or absence of a single individual in the dataset should not impact a differentially private analysis "too much." We formalize this below.

Definition 12 (Differential Privacy) *A randomized algorithm $\mathcal{M} : X^n \to \mathcal{R}$ is (ε, δ)-differentially private if for all measurable $T \subseteq \mathcal{R}$ and all neighboring datasets $S \sim S' \in X^n$, we have*

$$\Pr[\mathcal{M}(S) \in T] \leq e^{\varepsilon} \Pr[\mathcal{M}(S') \in T] + \delta.$$

In our analyses, it will actually be more useful to work with the notion of (zero-)concentrated differential privacy, which bounds higher moments of privacy loss than normal differential privacy.

Definition 13 (Zero Concentrated Differential Privacy (zCDP)) *A randomized algorithm $\mathcal{M} : X^n \to \mathcal{R}$ satisfies ρ-zCDP if for all neighboring datasets $S \sim S' \in X^n$ and all $\alpha > 1$, we have $D_\alpha(\mathcal{M}(S) \parallel \mathcal{M}(S')) \leq \rho\alpha$, where $D_\alpha(\cdot \parallel \cdot)$ denotes the Rényi divergence of order α.*

This second notion will often be more convenient to work with, because it tightly captures the privacy guarantee of Gaussian noise addition and of composition:

Lemma 14 (Tight Composition for zCDP, Bun and Steinke (2016)) *If $\mathcal{M}_1 : X^n \to \mathcal{R}_1$ satisfies ρ_1-zCDP, and $\mathcal{M}_2 : (X^n \times \mathcal{R}_1) \to \mathcal{R}_2$ satisfies ρ_2-zCDP, then the composition $\mathcal{M} : X^n \to \mathcal{R}_2$ defined by $\mathcal{M}(S) = \mathcal{M}_2(S, \mathcal{M}_1(S))$ satisfies $(\rho_1 + \rho_2)$-zCDP.*

zCDP can be converted into a guarantee of (ε, δ)-differential privacy.

Lemma 15 (zCDP $\Longrightarrow$ DP, Bun and Steinke (2016)) *Let $\mathcal{M} : X^n \to \mathcal{R}$ satisfy ρ-zCDP. Then for every $\delta > 0$, we have that $\mathcal{M}$ also satisfies (ε, δ)-differential privacy for $\varepsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$.*

The following lemma will let us bound Rényi divergence between related Gaussians:

Lemma 16 (Folklore, see Bun and Steinke (2016)) *Let $z, z' \in \mathbb{R}^d$, $\sigma \in \mathbb{R}$, and $\alpha \in [1, \infty)$. Then*

$$D_\alpha(\mathcal{N}(z, \sigma^2 I_d) \parallel \mathcal{N}(z', \sigma^2 I_d)) = \frac{\alpha \|z - z'\|_2^2}{2\sigma^2}.$$

Finally, ρ-zCDP is closed under post-processing, just like standard DP.

Lemma 17 (Post-processing zCDP, Bun and Steinke (2016)) *Let $\mathcal{M} : X^n \to R_1$ and $f : R_1 \to R_2$ be randomized algorithms. Suppose $\mathcal{M}$ satisfies ρ-zCDP. Define $\mathcal{M}' : X^n \to R_2$ by $\mathcal{M}'(x) = f(M(x))$. Then $\mathcal{M}'$ satisfies ρ-zCDP.*

3. Abstract Boosting (for Privacy)

Here we give sufficient conditions for private boosting, using a natural decomposition that applies to many boosting algorithms. In Appendix G, we use the same decomposition to give templates for noise-tolerant generalization and sample complexity bounds for private boosted classifiers that rely only on the algorithmic stability imposed by differential privacy. While boosting is amenable to generalization arguments using many different techniques (e.g., VC Theory, compression-based arguments, and margin bounds) stability-based arguments have not yet produced concrete sample bounds for boosting (Gao and Zhou, 2010). But when we impose the stringent algorithmic stability condition of differential privacy, a straightforward generalization argument follows. Furthermore, the sample bounds from DP alone are not much worse than those based on fat-shattering dimension for large-margin halfspaces (Appendix H).

3.1. Boosting Schemas

A boosting algorithm repeatedly calls a *weak* learning algorithm, aggregating the results to produce a final hypothesis that has good training error. Each call to the weak learner re-weights the samples so that samples predicted poorly by the hypothesis collection so far are given more "attention" (probability mass) by the weak learner in subsequent rounds. Thus boosting naturally decomposes into two[2] algorithmic parts: the weak learner `WkL` and the re-weighting strategy `NxM`.

Below, we describe boosting formally using a "helper function" to iterate weak learning and re-weighting. Crucially, we avoid iterating over any information regarding the intermediate weights; the entire state of our schema is a list of hypotheses. This makes it easy to apply a privacy-composition theorem to any boosting algorithm where `NxM` and `WkL` satisfy certain minimal conditions, elaborated later. Much of the complexity in the analysis of private boosting by Dwork et al. (2010) was due to carefully privatizing auxiliary information about sample weights; we avoid that issue entirely. So, many smooth boosting algorithms can be easily adapted to our framework.

We denote by $\mathcal{H}$ the hypotheses used by the weak learner, by S an i.i.d. sample from the target distribution D, by T the number of rounds, by $\mathfrak{M}$ the set of bounded measures over S, and by $\mathcal{D}(S)$ the set of distributions over S.

[2] Some boosting algorithms also have a *stopping rule;* instead of a fixed number of rounds T they terminate when the ensemble is "good enough." We give an appropriate schema in Appendix K and capture Servedio's `SmoothBoost`.

Algorithm 1 Boost. In: $S \in X^n,\ T \in \mathbb{N}$

```
$H \leftarrow \{\}$
for $t = 1$ to $T$ do
    $H \leftarrow \texttt{Iter}(S, H)$
end for
$\hat{f}(x) \leftarrow \frac{1}{T} \sum_{i=1}^{T} h_i(x)$
return $\operatorname{sgn}(\hat{f}(x))$
```

Algorithm 2 Iter. In: $S \in X^n,\ H \in \mathcal{H}^*$

```
$\mu \leftarrow \texttt{NxM}(S, H)$
$h \leftarrow \texttt{WkL}(S, \mu)$
return $H \cup \{h\}$
// Add $h$ to list of hypotheses
```

3.2. Ensuring Private Boosting

Under what circumstances will boosting algorithms using this schema guarantee differential privacy? Since we output (with minimal post-processing) a collection of hypotheses from the weak learner, it should at least be the case that the weak learning algorithm WkL is itself differentially private. In fact, we will need that the output distribution on hypotheses of a truly private weak learner does not vary too much if it is called with both similar *samples* and similar *distributional targets*.

Definition 18 (Private Weak Learning) *A weak learning algorithm* $\texttt{WkL} : S \times \mathcal{D}(S) \to \mathcal{H}$ *satisfies* (ρ, s)*-zCDP if for all neighboring samples* $S \sim S' \in (\mathcal{X}^n \times \{\pm 1\})$*, all* $\alpha > 1$*, and any pair of distributions* $\hat{\mu}, \hat{\mu}'$ *on* X *such that* $\Delta(\hat{\mu}, \hat{\mu}') < s$*, we have:*

$$D_\alpha(\texttt{WkL}(S, \hat{\mu}) \parallel \texttt{WkL}(S', \hat{\mu}')) \leq \rho\alpha$$

This makes it natural to demand that neighboring samples induce similar measures. Formally:

Definition 19 (ζ-Slick Measure Production) *A measure production algorithm* $\texttt{NxM} : S \times \mathcal{H} \to \mathfrak{M}$ *is called* ζ*-slick if, for all neighboring samples* $S \sim S' \in (\mathcal{X}^n \times \{\pm 1\})$ *and for all sequences of hypotheses* $H \in \mathcal{H}^*$*, letting* $\hat{\mu}$ *and* $\hat{\mu}'$ *be the distributions induced by* $\texttt{NxM}(S, H)$ *and* $\texttt{NxM}(S', H)$ *respectively, we have:*

$$\Delta(\hat{\mu}, \hat{\mu}') \leq \zeta$$

It is immediate that a single run of Iter is private if it uses NxM and WkL procedures that are appropriately slick and private, respectively. Suppose WkL is (ρ_W, ζ)-zCDP and NxM is ζ-slick. By composition, Iter run using these procedures is ρ_W-zCDP. Finally, observe that Boost paired with a private weak learner and slick measure production is is $T\rho_W$-zCDP, because the algorithm simply composes T calls to Iter and then post-processes the result.

4. Concrete Boosting via Lazy Bregman Projection

We instantiate the framework above. This requires a "Next Measure" routine (LB-NxM, Algorithm 3) a Boosting Theorem (Theorem 20) and a slickness bound (Lemma 21).

4.1. Measure Production Using Lazy Dense Multiplicative Weights

Our re-weighting strategy combines multiplicative weights with Bregman projections. In each round, we compute the collective margin on each example. Then, we multiplicative-weight the examples according to error: examples predicted poorly receive more weight. Finally, to ensure that

Algorithm 3 `LB-NxM(`κ, λ`)` : Lazy-Bregman Next Measure

Parameters: $\kappa \in (0, 1)$, desired density of output measures; $\lambda \in (0, 1)$, learning rate
Input: S, the sample; $H = \{h_1, \ldots, h_t\}$, a sequence of hypotheses
Output: A measure over $[n]$, $n = |S|$

```
$\mu_1(i) \leftarrow \kappa \;\; \forall i \in [n]$   /* Initial measure is uniformly $\kappa$ */
for $j \in [t]$ do
    $\ell_j(x_i) \leftarrow 1 - \frac{1}{2}|h_j(x_i) - y_i| \;\; \forall i \in [n]$   /* Compute error of each hypothesis */
end for
$\tilde{\mu}_{t+1}(i) \leftarrow e^{-\lambda \sum_{j=1}^{t} \ell_j(x_i)} \mu_1(i) \;\; \forall i \in [n]$
$\mu_{t+1} \leftarrow \Pi_\Gamma(\tilde{\mu}_{t+1})$
return $\hat{\mu}_{t+1}$
```

no example receives too much weight, we Bregman-project the resulting measure into the space Γ of κ-dense measures. We call this strategy "lazy" because projection happens only *once* per round.

`LB-NxM` is typed correctly for substitution into the `Boost` algorithm above; the measure is computed using *only* a sample and current list of hypotheses. Thus, `LazyBregBoost` = `Boost(LB-NxM)` admits a simple privacy analysis as in Section 3.2.

4.2. Boosting Theorem for Lazy Bregman Projection

Given a weak learner that beats random guessing, running `LazyBregBoost` yields low training error after a bounded number of rounds; we prove this in Appendix I. Our argument adapts the well-known reduction from boosting to iterated play of zero-sum games (Freund and Schapire, 1996) for hypotheses with real-valued outputs. For completeness, we also give a self-contained analysis of the iterated-play strategy corresponding to `LB-NxM` in Appendix J. Similar strategies are used by other differentially-private algorithms (Hsu et al., 2013, 2014) and their properties are known to follow from regret bounds for lazy projected mirror descent (Shalev-Shwartz, 2012; Hazan, 2016). However, to our knowledge an explicit proof for the variant above does not appear in the literature; so we include one in Appendix J for completeness. Overall, we have the follwing:

Theorem 20 (Lazy Bregman Round-Bound) *Suppose we run* `Boost` *with* `LB-NxM(`$\kappa, \gamma/4$`)` *on a sample* $S \subset \mathcal{X} \times \{\pm 1\}$ *using any real-valued weak learner with advantage* γ *for* $T \geq \frac{16 \log(1/\kappa)}{\gamma^2}$ *rounds. Let* $H : \mathcal{X} \to [-1, 1]$ *denote the final, aggregated hypothesis. The process has:*

Good Margin: *H mostly agrees with the labels of S.*

$$\Pr_{(x,y)\sim S}[yH(x) \leq \gamma] \leq \kappa$$

Smoothness: *Every distribution $\hat{\mu}_t$ supplied to the weak learner has $\hat{\mu}_t(i) \leq \frac{1}{\kappa n} \; \forall i$*

4.3. Slickness Bound for Lazy Bregman Projection

`LB-NxM` is "lazy" in the sense that Bregman projection occurs only once per round, after all the multiplicative updates. The projection step is *not* interleaved between multiplicative updates. This is necessary to enforce slickness, which we require for privacy as outlined in Section 3.2.

Lemma 21 (Lazy Bregman Slickness) *The dense measure update rule* `LB-NxM` *(Algorithm 3) is ζ-slick for $\zeta = 1/\kappa n$.*

Proof [sketch, see Appendix D.] Observe that, when run on neighboring datasets $S \sim S'$, the unprojected measures $\tilde{\mu}$ and $\tilde{\mu}'$ produced by `LB-NxM` can differ on exactly one element: the element z witnessing the single difference between S and S'.

Since the Bregman-projected and normalized distributions are κ-smooth, they cannot allocate too much mass to each individual element of the domain. Therefore, even the worst-case difference between $\mu(z)$ and $\mu'(z)$ is not enough to shift $\Delta(\hat{\mu}, \hat{\mu}')$ by too much, after a *single* projection. ■

5. Application: Learning Halfspaces with a Margin

5.1. Learning Settings

We first assume realizability by a large-margin halfspace. Let u be an unknown unit vector in $\mathbb{R}^d$, and let D be a distribution over examples from the ℓ_2 unit ball $B_d(1) \subset \mathbb{R}^d$. Further suppose that D is τ-good for u, meaning $|u \cdot x| \geq \tau$ for all x in the support of D. A PAC learner is given access to n i.i.d. labeled samples from D, honestly labeled by u.

A noise-tolerant learner is given access to a *label noise* example oracle with noise rate η, which behaves as follows. With probability $1 - \eta$, the oracle returns a clean example $(x, \operatorname{sgn}(u \cdot x))$ for $x \sim D$. With probability η, the oracle returns an example with the label flipped: $(x, -\operatorname{sgn}(u \cdot x))$ for $x \sim D$. Given access to the noisy example oracle, the goal of a leaner is to output a hypothesis $h : B_d \to \{-1, 1\}$ which α-approximates u under D, i.e., $\Pr_{x \sim D}[h(x) \neq \operatorname{sgn}(u \cdot x)] \leq \alpha$ (Angluin and Laird, 1987).

Servedio (2003) showed that smooth boosting can be used to solve this learning problem under the (more demanding) *malicious noise* rate $\eta = O(\alpha\tau)$ using sample complexity $n = \tilde{O}(1/(\tau\alpha)^2)$. We apply the Gaussian mechanism to his weak learner to construct a differentially private weak learner, and then boost it while preserving privacy. Our (best) sample complexity bounds then follow by appealing to the fat-shattering dimension of bounded-norm halfspaces in Section 5.4. Slightly worse bounds proved using only differential privacy are derived in Appendix H.

5.2. Weak Halfspace Learner: Centering with Noise

We apply the Gaussian mechanism to Servedio's weak learner to obtain $\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ which outputs $h(x) = \hat{z} \cdot x$ for

$$\hat{z} = \sum_{i=1}^{n} \hat{\mu}(j) \cdot y_i \cdot x_i + \nu,$$

with noise $\nu \sim \mathcal{N}(0, \sigma^2 I_d)$. Our advantage bound (proved in Appendix E) trades off against privacy.

Theorem 22 (Private Weak Halfspace Learner) *Let $\hat{\mu}$ be a distribution over $[n]$ such that $L_\infty(\hat{\mu}) \leq 1/\kappa n$. Suppose that at most ηn examples in S do not satisfy the condition $y_i \cdot (u \cdot x_i) \geq \tau$ for $\eta \leq \kappa\tau/4$. Then we have:*

1. ***Privacy:*** *$\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ satisfies (ρ, s)-zCDP for $\rho = \frac{2(1/\kappa n + s)^2}{\sigma^2}$.*

2. ***Advantage:*** *There is a constant c such that for any $\xi > 0$, $\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ returns a hypothesis $h : B_d \to [-1, 1]$ that, with probability at least $1 - \xi$, has advantage at least $\tau/4 - c\sigma\sqrt{\log(1/\xi)}$ under $\hat{\mu}$.*

5.3. Strong Halfspace Learner: Boosting

Putting all the pieces together, we run `Boost` using the private weak halfspace learner (Theorem 22) and lazy-Bregman measures (Theorem 20). Via the composition theorem for differential privacy, we get a privacy guarantee for the terminal hypothesis as outlined in Section 3.2. Finally, we use the fat shattering dimension to ensure that this hypothesis generalizes.

Algorithm 4 Strong Halfspace Learner, via Boosting (`HS-StL`)

```
Input: Sample: $S$; Parameters: $(\alpha, \beta)$-PAC, $(\varepsilon, \delta)$-DP, $\tau$-margin
Output: A hypothesis $H$
    $\sigma \leftarrow \tau/8c\sqrt{\log\left(\frac{3072\log(1/\kappa)}{\beta\tau^2}\right)}$
    $T \leftarrow 1024\log(1/\kappa)/\tau^2$
    $H \leftarrow$ Boost run with LB-NxM$(\kappa := (\alpha/4), \lambda := (\tau/8))$ and $\widehat{\mathrm{WL}}(\cdot, \cdot, \sigma)$ for $T$ rounds
```

5.4. Generalization via fat-shattering dimension.

Following the analysis of Servedio (2003), we can show that with high probability the hypothesis output by our halfspace learner will generalize, even for a sample drawn from a distribution with random classification noise at rate $O(\alpha\tau)$. The proof of generalization goes by way of fat-shattering dimension. Using an argument nearly identical to that of Servedio (2000), we can bound the fat-shattering dimension of our hypothesis class. This bound, along with the guarantee of Theorem 20 that our final hypothesis will have good margin on a large fraction of training examples, allows us to apply the following generalization theorem of Bartlett and Shawe-Taylor, which bounds the generalization error of the final hypothesis.

Theorem 23 (Bartlett and Shawe-Taylor (1998)) *Let $\mathcal{H}$ be a family of real-valued hypotheses over some domain $\mathcal{X}$, let D be a distribution over labeled examples $\mathcal{X} \times \{-1, 1\}$. Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a sequence of labeled examples from D, and let $h(x) = \mathrm{sgn}(H(x))$ for some $H \in \mathcal{H}$. If h has margin less than γ on at most k examples in S, then with probability at least $1 - \delta$ we have that*

$$\Pr_{(x,y)\sim D}[h(x) \neq y] \leq \frac{k}{n} + \sqrt{\frac{2d}{n}\ln(34en/d)\log(578n) + \ln(4/\delta)}$$

where $d = fat_F(\gamma/16)$ is the fat-shattering dimension of $\mathcal{H}$ with margin $\gamma/16$.

In order to meaningfully apply the above theorem, we will need to bound the fat-shattering dimension of our hypothesis class $\mathcal{H}$. Our bound (proved in Appendix F) follows from the analysis of Servedio (2000), but given that our hypothesis class is not exactly that analyzed in Servedio (2000), the bound holds only when the noise added to the hypotheses at each round of boosting does not increase the ℓ_2 norm of the final hypothesis by too much.

Lemma 24 (Fuzzy Halfspace Fat Shattering Dimension) *With probability* $1 - \frac{\beta}{3}$*, after* $T = \frac{1024 \log(1/\kappa)}{\tau^2}$ *rounds of boosting, Algorithm 5.3 outputs a hypothesis in a class with fat-shattering dimension* $fat_{\mathcal{H}}(\gamma) \leq 4/\gamma^2$.

With the above bound on fat-shattering dimension, we may prove the following properties of `HS-StL`.

Theorem 25 (Private Learning of Halfspaces) *The* `HS-StL` *procedure, is a* (ε, δ)*-Differentially Private* (α, β)*-strong PAC learner for* τ*-margin halfspaces, tolerating random classification noise at rate* $O(\alpha\tau)$*, with sample complexity*

$$n = \Omega\left(\underbrace{\frac{\sqrt{\log(1/\alpha)\log(1/\delta)\log(\log(1/\alpha)/\beta\tau^2)}}{\varepsilon\alpha\tau^2}}_{\textit{privacy}} + \underbrace{\frac{\log(1/\tau\alpha)\log(1/\alpha)}{\alpha^2\tau^2} + \frac{\log(1/\beta)}{\alpha\tau}}_{\textit{accuracy}} \right)$$

Proof [sketch, see Appendix C] We begin by calculating the total zCDP guarantee of `HS-StL`. First, by the privacy bound for $\widehat{\mathrm{WL}}$ (Theorem 22), we know that a single iteration of `Boost` is ρ-zCDP for $\rho = \frac{8}{(\kappa n \sigma)^2}$. Furthermore, by tight composition for zCDP (Lemma 14) and our setting of T, `HS-StL` is ρ_T-zCDP where:

$$\rho_T = O\left(\frac{\log(1/\kappa)}{(\kappa n \sigma \tau)^2}\right).$$

Denote by ε and δ the parameters of approximate differential privacy at the final round T of `HS-StL`. Now we convert from zero-concentrated to approximate differential privacy, via Lemma 15: for all $\delta > 0$, if $\varepsilon > 3\sqrt{\rho_T \log(1/\delta)}$, then `HS-StL` is (ε, δ)-DP. So, for a given target ε and δ, taking

$$n \in O\left(\frac{\sqrt{\log(1/\kappa)\log(1/\delta)\log(\log(1/\kappa)/\beta\tau)}}{\varepsilon\kappa\tau^2}\right)$$

will ensure the desired privacy.

If no "bad" events that ruin either training error or generalization occur (see Appendix C for details), it remains to show that we can achieve accuracy α. From Lemma 39, we have that H will have margin $\gamma = \tau/8$ on all but a κ fraction of the examples, some of which may have been corrupted. We assume the worst case – that H is correct on all corrupted examples. Then if we set $\kappa = \alpha/4$ and take

$$n \in O\left(\frac{\log(1/\alpha\gamma)\log(1/\alpha)}{\alpha^2\tau^2}\right),$$

we can apply Theorem 23 to conclude that

$$\Pr_{S \sim D^n}\left[\Pr_{(x,y)\sim D}[H(x) \neq y] < \alpha\right] \geq 1 - \beta.$$

■

Acknowledgments

We thank Russell Impagliazzo for questions that prompted this work and many helpful discussions. MB is supported by NSF grant CCF-1947889 and part of this work was done at the Simons Institute for the Theory of Computing, supported by a Google Research Fellowship. MLC is supported by a PIMS postdoctoral fellowship. JS is supported by the Simons Foundation, NSF grant CCF-1936703, and NSF grant CCF-1909634.

References

Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987. doi: 10.1007/BF00116829. URL https://doi.org/10.1007/BF00116829.

Boaz Barak, Moritz Hardt, and Satyen 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, New York, NY, USA, January 4-6, 2009*, pages 1193–1200. SIAM, 2009a. URL http://dl.acm.org/citation.cfm?id=1496770.1496899.

Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2009, pages 1193–1200. SIAM, 2009b.

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers, 1998.

Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1046–1059. ACM, 2016. doi: 10.1145/2897518.2897566. URL https://doi.org/10.1145/2897518.2897566.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658, 2016. doi: 10.1007/978-3-662-53641-4_24. URL https://doi.org/10.1007/978-3-662-53641-4_24.

Carlos Domingo and Osamu Watanabe. Madaboost: A modification of adaboost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, COLT '00, pages 180–189. Morgan Kaufmann, 2000.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 371–380, 2009.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60. IEEE Computer Society, 2010.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In Avrim Blum and Michael J. Kearns, editors, *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996*, pages 325–332. ACM, 1996. doi: 10.1145/238061.238163. URL https://doi.org/10.1145/238061.238163.

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

Wei Gao and Zhi-Hua Zhou. Approximation stability and boosting. In Marcus Hutter, Frank Stephan, Vladimir Vovk, and Thomas Zeugmann, editors, *Algorithmic Learning Theory, 21st International Conference, ALT 2010, Canberra, Australia, October 6-8, 2010. Proceedings*, volume 6331 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2010. doi: 10.1007/978-3-642-16108-7_9. URL https://doi.org/10.1007/978-3-642-16108-7_9.

Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3-4):157–325, 2016. doi: 10.1561/2400000013. URL https://doi.org/10.1561/2400000013.

Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 341–350. ACM, 2013. doi: 10.1145/2488608.2488651. URL https://doi.org/10.1145/2488608.2488651.

Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. Privately solving linear programs. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 612–624. Springer, 2014. doi: 10.1007/978-3-662-43948-7_51. URL https://doi.org/10.1007/978-3-662-43948-7_51.

Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pages 538–545. IEEE Computer Society, 1995.

Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees. *CoRR*, abs/1909.03577, 2019. URL http://arxiv.org/abs/1909.03577.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? 40(3):793–826, 2011.

Huy L. Nguyễn, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning halfspaces. 2019.

Alexander Rakhlin. Lecture notes on online learning (draft), 2009. URL http://www.mit.edu/~rakhlin/papers/online_learning.pdf.

Rocco A. Servedio. PAC analogues of perceptron and winnow via boosting the margin. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT 2000), June 28 - July 1, 2000, Palo Alto, California, USA*, pages 148–157, 2000.

Rocco A. Servedio. Smooth boosting and learning with malicious noise. *J. Mach. Learn. Res.*, 4: 633–648, 2003. URL http://jmlr.org/papers/v4/servedio03a.html.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. doi: 10.1561/2200000018. URL https://doi.org/10.1561/2200000018.

Leslie G. Valiant. A theory of the learnable (cacm). *Commun. ACM*, 27(11):1134–1142, 1984. doi: 10.1145/1968.1972. URL http://doi.acm.org/10.1145/1968.1972.

Leslie G. Valiant. Learning disjunction of conjunctions. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, IJCAI ’85, pages 560–566. Morgan Kaufmann, 1985.

Appendix A. Glossary of Symbols

n Sample size

$\mathcal{M}$ A randomized mechanism.

$\hat{\alpha}$ Training error. Only in Appendices.

α Final desired accuracy of output hypothesis, as in (α, β)-PAC learning

β Probability of catastrophic learning failure, as in (α, β)-PAC learning

ε Final desired privacy of output hypothesis, as in (ε, δ)-DP

δ Final desired "approximation" of privacy, as in (ε, δ)-DP

η Rate of random classification noise

τ Margin of underlying target halfspace.

ζ Slickness parameter.

d Dimension of the input examples from $\mathbb{R}^d = X$

γ Advantage of Weak Learner

ν Gaussian random variable denoting noise added to Weak Learner

σ Magnitude of Gaussian noise added to Weak Learner

κ Density parameter of LazyBregBoost

λ Learning rate of Dense Multiplicative Weights

θ Margin induced by "Optimal" Boosting (Pythia)

X A domain.

$\mathcal{X}$ Domain of examples, specifically.

μ A bounded measure.

$\hat{\mu}$ A normalized bounded measure; a distribution.

$\tilde{\mu}$ A bounded measure that has *not* been Bregman projected.

$\Delta(Z, Y)$ Total variation distance between Z and Y

M A two-player zero-sum game. Only in Appendices.

Appendix B. Table of Contents

Contents

Appendix C. Proof of generalization via fat-shattering dimension

We recall and prove Theorem 26.

Theorem 26 (Private Learning of Halfspaces) *The* `HS-StL` *procedure, is a (ε, δ)-Differentially Private (α, β)-strong PAC learner for τ-margin halfspaces, tolerating random classification noise at rate $O(\alpha\tau)$, with sample complexity*

$$n = \Omega\left(\underbrace{\frac{\sqrt{\log(1/\alpha)\log(1/\delta)\log(\log(1/\alpha)/\beta\tau^2)}}{\varepsilon\alpha\tau^2}}_{\textit{privacy}} + \underbrace{\frac{\log(1/\tau\alpha)\log(1/\alpha)}{\alpha^2\tau^2} + \frac{\log(1/\beta)}{\alpha\tau}}_{\textit{accuracy}} \right)$$

Proof We begin by calculating the cumulative zCDP guarantee of `HS-StL`. First, by the privacy bound for $\widehat{\mathrm{WL}}$ (Theorem 22), we know that a single iteration of `Boost` is ρ-zCDP for $\rho = \frac{8}{(\kappa n\sigma)^2}$. Furthermore, by tight composition for zCDP (Lemma 14) and our setting of T, `HS-StL` is ρ_T-zCDP where:

$$\rho_T = O\left(\frac{\log(1/\kappa)}{(\kappa n \sigma\tau)^2}\right).$$

Denote by ε and δ the parameters of approximate differential privacy at the final round T of `HS-StL`. Now we convert from zero-concentrated to approximate differential privacy, via Lemma 15: for all $\delta > 0$, if $\varepsilon > 3\sqrt{\rho_T \log(1/\delta)}$, then `HS-StL` is (ε, δ)-DP. So, for a given target ε and δ, taking

$$n \in O\left(\frac{\sqrt{\log(1/\kappa)\log(1/\delta)\log(\log(1/\kappa)/\beta\tau)}}{\varepsilon\kappa\tau^2}\right)$$

will ensure the desired privacy.

We now turn to bounding the probability of events that could destroy good training error.

Too Many Corrupted Samples. Our proof of $\widehat{\mathrm{WL}}$'s advantage required that fewer than $\kappa\tau n/4$ examples are corrupted. At noise rate $\eta \leq \kappa\tau/8$, we may use a Chernoff bound to argue that the probability of exceeding this number of corrupted samples is at most $\beta/3$, by taking $n > \frac{24\log(3/\beta)}{\kappa\tau}$.

Gaussian Mechanism Destroys Utility. The Gaussian noise injected to ensure privacy could destroy utility for a round of boosting. Our setting of σ simplifies the advantage of $\widehat{\mathrm{WL}}$ to $\gamma(\tau, \sigma) = \tau/8$ with all but probability $\xi = \frac{\beta\tau^2}{3072\log(1/\kappa)}$. Then we have that with probability $(1-\xi)^T \geq 1 - \frac{\beta}{3}$, every hypothesis output by $\widehat{\mathrm{WL}}$ satisfies the advantage bound $\gamma \geq \tau/8$. Therefore, by Theorem 20, `HS-StL` only fails to produce a hypothesis with training error less than κ with probability $\beta/3$.

We now consider events that cause generalization to fail.

Final hypothesis $H \notin \mathcal{H}$. The Gaussian noise added to ensure privacy could cause the final hypothesis H to fall outside the class $\mathcal{H} = \{f(x) = z \cdot x : \|z\|_2 \leq 2\}$, for which we have a fat-shattering dimension bound. The probability of this event, however, is already accounted for by the probability that the Gaussian Mechanism destroys the weak learner's utility, as both failures follow from the Gaussian noise exceeding some ℓ_2 bound. The failures that affect utility are a superset of those that affect the fat-shattering bound, and so the $\beta/3$ probability of the former subsumes the probability of the latter.

Failure internal to generalization theorem. Theorem 23 gives a generalization guarantee that holds only with some probability. We denote the probability of this occurrence by β_1.

If none of these failures occur, it remains to show that we can achieve accuracy α. From Lemma 39, we have that H will have margin $\gamma = \tau/8$ on all but a κ fraction of the examples, some of which may have been corrupted. We assume the worst case – that H is correct on all corrupted examples. We have already conditioned on the event that fewer than $\kappa\tau n/4$ examples have been corrupted, and so we may then conclude that H has margin less than γ on at most a 2κ fraction of the uncorrupted examples. Then if we set $\kappa = \alpha/4$ and take

$$n \in O\left(\frac{\log(1/\alpha\gamma)\log(1/\alpha)}{\alpha^2\tau^2}\right),$$

then so long as $e^{-\alpha^2} < \beta_1 < \beta/3$, we can apply Theorem 23 to conclude that

$$\Pr_{S\sim D^n}\left[\Pr_{(x,y)\sim D}[H(x) \neq y] < \alpha\right] \geq 1 - \beta.$$

■

Appendix D. Lazy Bregmans are Slick

We recall and prove Lemma 21.

Lemma 21 (Lazy Bregman Slickness) *The dense measure update rule* `LB-NxM` *(Algorithm 3) is ζ-slick for $\zeta = 1/\kappa n$.*

Proof Let $\tilde{\mu}, \tilde{\mu}'$ be the unprojected measures produced at the end of the outermost loop of `NxM`, when `NxM` is run with the sequence of hypotheses $H = \{h_1, \ldots, h_T\}$, and on neighboring datasets $S \sim S'$. Let i be the index at which S and S' differ, and note that $\tilde{\mu}(j) = \tilde{\mu}'(j)$ for all $j \neq i$.

Let $\tilde{\mu}_0$ denote the measure with $\tilde{\mu}_0(j) = \tilde{\mu}(j) = \tilde{\mu}'(j)$ for all $j \neq i$, and $\tilde{\mu}_0(i) = 0$. Take Γ to be the space of κ-dense measures, and let $\mu_0 = \Pi_\kappa\tilde{\mu}_0$ and $\mu = \Pi_\kappa\tilde{\mu}$ denote the respective projected measures. We will show that $SD(\hat{\mu}_0, \hat{\mu}) \leq 1/\kappa n$, which is enough to prove the claim by the triangle inequality. (Note that $|\mu_0| = |\mu| = \kappa n$, which follows from Lemma 9 and the observation that $|\tilde{\mu}_0| \leq |\tilde{\mu}| \leq \kappa n$. Moreover, $\mu_0(j) \geq \mu(j)$ for every $j \neq i$.)

We calculate

$$\begin{aligned}
\sum_{j=1}^{n} |\mu_0(j) - \mu(j)| &= |\mu(i)| + \sum_{j\neq i} |\mu_0(j) - \mu(j)| \\
&\leq 1 + \sum_{j\neq i} \mu_0(j) - \mu(j) \\
&= 1 + |\mu_0| - (|\mu| - \mu(i)) \\
&\leq 1 + |\mu_0| - |\mu| + 1 \\
&= 2,
\end{aligned}$$

since μ and μ_0 have density κ. Hence,

$$\begin{aligned}\Delta(\hat{\mu}, \hat{\mu_0}) &= \frac{1}{2}\sum_{i=1}^{n}\left|\frac{\mu(i)}{|\mu|} - \frac{\mu_0(i)}{|\mu_0|}\right| \\ &= \frac{1}{2\kappa n}\sum_{i=1}^{n}|\mu(i) - \mu_0(i)| \\ &\leq \frac{1}{\kappa n}.\end{aligned}$$

■

Appendix E. Private Weak Halfspace Learner: Full Analysis

The noise-tolerant weak learner for halfspaces was $\mathrm{WL}(S, \hat{\mu})$ which outputs the hypothesis $h(x) = z \cdot x$ where

$$z = \sum_{i=1}^{n} \hat{\mu}(j) \cdot y_i \cdot x_i.$$

The accuracy of this learner is given by the following theorem:

Theorem 27 (Servedio (2003)) *Let $\hat{\mu}$ be a distribution over $[n]$ such that $L_\infty(\hat{\mu}) \leq 1/\kappa n$. Suppose that at most ηn examples in S do not satisfy the condition $y_i \cdot (u \cdot x_i) \geq \tau$ for $\eta \leq \kappa\tau/4$. Then $\mathrm{WL}(S, \hat{\mu})$ described above returns a hypothesis $h : B_d \to [-1, 1]$ with advantage at least $\tau/4$ under $\hat{\mu}$.*

We apply the Gaussian mechanism to Servedio's weak learner to obtain $\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ which outputs $h(x) = \hat{z} \cdot x$ for

$$\hat{z} = \sum_{i=1}^{n} \hat{\mu}(j) \cdot y_i \cdot x_i + \nu,$$

with noise $\nu \sim \mathcal{N}(0, \sigma^2 I_d)$. We get a similar advantage bound (proved in Appendix E), now trading off with privacy.

We recall and prove Theorem 22.

Theorem 22 (Private Weak Halfspace Learner) *Let $\hat{\mu}$ be a distribution over $[n]$ such that $L_\infty(\hat{\mu}) \leq 1/\kappa n$. Suppose that at most ηn examples in S do not satisfy the condition $y_i \cdot (u \cdot x_i) \geq \tau$ for $\eta \leq \kappa\tau/4$. Then we have:*

1. ***Privacy:*** *$\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ satisfies (ρ, s)-zCDP for $\rho = \frac{2(1/\kappa n + s)^2}{\sigma^2}$.*

2. ***Advantage:*** *There is a constant c such that for any $\xi > 0$, $\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ returns a hypothesis $h : B_d \to [-1, 1]$ that, with probability at least $1 - \xi$, has advantage at least $\tau/4 - c\sigma\sqrt{\log(1/\xi)}$ under $\hat{\mu}$.*

Proof We begin with the proof of privacy. Let $\hat{\mu}_1, \hat{\mu}_2$ be κ-smooth distributions over $[n]$ with statistical distance $\Delta(\hat{\mu}_1, \hat{\mu}_2) \leq s$. Let $S \sim S'$ be neighboring datasets with $\{(x_i, y_i)\} = S \setminus S'$ and $\{(x'_i, y'_i)\} = S' \setminus S$. Then we have

$$\begin{aligned}
\|\hat{z}_{S,\hat{\mu}_1} - \hat{z}_{S',\hat{\mu}_2}\|_2 &= \|\hat{\mu}_1(i) y_i \cdot x_i - \hat{\mu}_2(i) y'_i \cdot x'_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} (\hat{\mu}_1(j) - \hat{\mu}_2(j)) y_j \cdot x_j \|_2 \\
&\leq \|\hat{\mu}_1(i) y_i \cdot x_i\|_2 + \|\hat{\mu}_2(i) y'_i \cdot x'_i\|_2 + \sum_{\substack{j=1 \\ j\neq i}}^{n} \|(\hat{\mu}_1(j) - \hat{\mu}_2(j)) y_j \cdot x_j\|_2 \\
&= \hat{\mu}_1(i) + \hat{\mu}_2(i) + \sum_{\substack{j=1 \\ j\neq i}}^{n} |(\hat{\mu}_1(j) - \hat{\mu}_2(j))| \\
&\leq 2\hat{\mu}_2(i) + \sum_{j=1}^{n} |\hat{\mu}_1(j) - \hat{\mu}_2(j)| \\
&\leq 2(1/\kappa n + s).
\end{aligned}$$

Then Lemma 16 gives us that

$$D_\alpha\left(\widehat{\mathrm{WL}}(S, \hat{\mu}_1, \sigma) \,\middle\|\, \widehat{\mathrm{WL}}(S', \hat{\mu}_2, \sigma)\right) \leq \frac{2\alpha(1/\kappa n + s)^2}{\sigma^2}$$

and therefore $\widehat{\mathrm{WL}}(S, \hat{\mu}_1, \sigma)$ satisfies (ρ, s)-zCDP for $\rho = \frac{2(1/\kappa n+s)^2}{\sigma^2}$.

Building on Servedio's result, we now give the advantage lower bound. Servedio's argument shows that the advantage of $\widehat{\mathrm{WL}}(S, \hat{\mu}, \sigma)$ is at least $\hat{z} \cdot u/2 = z \cdot u/2 + \nu \cdot u/2$. Since ν is a spherical Gaussian and u is a unit vector, we have that for any $\xi > 0$,

$$\Pr[|\nu \cdot u| \geq c\sigma\sqrt{\log(1/\xi)}] \leq \xi.$$

■

Appendix F. Fuzzy Halfspaces have Bounded Fat-Shattering Dimension

We recall and prove Lemma 24.

Lemma 24 (Fuzzy Halfspace Fat Shattering Dimension) *With probability $1-\frac{\beta}{3}$, after $T = \frac{1024\log(1/\kappa)}{\tau^2}$ rounds of boosting, Algorithm 5.3 outputs a hypothesis in a class with fat-shattering dimension $fat_{\mathcal{H}}(\gamma) \leq 4/\gamma^2$.*

This follows from the lemmas below due to Servedio, Bartlett, and Shawe-Taylor.

Lemma 28 (Servedio (2000)) *If the set $\{x_1, \ldots, x_n\}$ is γ-shattered by $\mathcal{H} = \{f(x) = z \cdot x : \|z\|_2 \leq 2\}$, then every $b \in \{-1, 1\}^n$ satisfies*

$$\Big\| \sum_{i=1}^{n} b_i x_i \Big\|_2 \geq \gamma n/2.$$

Lemma 29 (Bartlett and Shawe-Taylor (1998)) *For any set $\{x_1, \dots, x_n\}$ with each $x_i \in \mathbb{R}^d$ and $\|x_i\|_2 \leq 1$, then there is some $b \in \{-1, 1\}^n$ such that $\| \sum_{i=1}^n b_i x_i \|_2 \leq \sqrt{n}$.*

Proof We begin by showing that, with high probability, the hypothesis output by Algorithm 5.3 is in the class $\mathcal{H} = \{f(x) = z \cdot x : \|z\|_2 \leq 2\}$. To bound the ℓ_2 norm of z, we observe that

$$\|z\|_2 = \frac{1}{T} \Big\| \sum_{t=1}^{T} \hat{z}_t \Big\|_2 \leq \frac{1}{T} \sum_{t=1}^{T} \Big\| \sum_{i=1}^{n} \hat{\mu}_t(i) y_i x_i \Big\|_2 + \|\nu_t\|_2 = 1 + \frac{1}{T} \sum_{t=1}^{T} \|\nu_t\|_2$$

where $\hat{z}_t$ denotes the weak learner hypothesis at round t of boosting, and ν_t denotes the Gaussian vector added to the hypothesis at round t. Letting $\sigma = \tau/8c\sqrt{\log\left(\frac{3072\log(1/\kappa)}{\beta\tau^2}\right)}$ for a constant c, it follows that with probability at least $1 - \frac{\beta\tau^2}{3072\log(1/\kappa)} = 1 - \frac{\beta}{3T}$, a given ν_t has $\|\nu_t\|_2 \leq \tau/8 < 1$, and therefore with probability $(1 - \frac{\beta}{3T})^T \geq (1 - \frac{\beta}{3})$, $\frac{1}{T}\sum_{t=1}^{T} \|\nu_t\|_2 < 1$, and so $\|z\|_2 \leq 2$. Therefore with all but probability $\beta/3$, the hypothesis output by Algorithm 5.3 is in the class $\mathcal{H}$.

From Lemma 28, it cannot be the case that a set $\{x_1, \dots, x_n\}$ is γ-shattered by $\mathcal{H}$ if there exists a $b \in \{-1, 1\}^n$ such that

$$\Big\| \sum_{i=1}^{n} b_i x_i \Big\|_2 < \gamma n/2.$$

At the same time, it follows from Lemma 29 that if $n > 4/\gamma^2$, such a $b \in \{-1, 1\}^n$ must exist. Therefore the fat-shattering dimension of $\mathcal{H}$ at margin γ is $fat_{\mathcal{H}}(\gamma) \leq 4/\gamma^2$. Since our final hypothesis is in $\mathcal{H}$ with probability $1 - \beta/3$, our claim holds. ■

Appendix G. Abstract Boosting (for Generalization)

Here we describe a proof template that combines private weak learning with private boosting to obtain noise-tolerant generalization guarantees and sample complexity bounds. First, we review the generalization properties of differential privacy.

G.1. Generalization and Differential Privacy

We now state the generalization properties of differentially private algorithms that select statistical queries, which count the fraction of examples satisfying a predicate. Let X be an underlying population, and denote by D a distribution over X.

Definition 30 (Statistical Queries) *A statistical query q asks for the expectation of some function on random draws from the underlying population. More formally, let $q : X \to [0, 1]$ and then define the statistical query* based on *q (abusing notation) as the following, on a sample $S \in X^n$ and the population, respectively:*

$$q(S) = \frac{1}{|S|} \sum_{x \in S} q(x) \qquad \textit{and} \qquad q(D) = \mathop{\mathbb{E}}_{x \sim D} [q(x)]$$

In the case of statistical queries, dependence of accuracy on the sample size is good enough to obtain interesting sample complexity bounds from privacy alone. These transfer theorems have recently been improved, bringing "differential privacy implies generalization" closer to practical utility by decreasing the very large constants from prior work (Jung et al., 2019). As our setting is asymptotic, we employ a very convienent (earlier) transfer lemma of Bassily et al. (2016).

Theorem 31 (Privacy $\implies$ Generalization of Statistical Queries, Bassily et al. (2016)) *Let $0 < \varepsilon < 1/3$, let $0 < \delta < \varepsilon/4$ and let $n \geq \log(4\varepsilon/\delta)/\varepsilon^2$. Let $\mathcal{M} : X^n \to Q$ be (ε, δ)-differentially private, where Q is the set of statistical queries $q : X \to \mathbb{R}$. Let D be a distribution over X, let $S \leftarrow_{\scriptscriptstyle R} D^n$ be an i.i.d. sample of n draws from D, and let $q \leftarrow_{\scriptscriptstyle R} \mathcal{M}(S)$. Then:*

$$\Pr_{q,S}\left[|q(S) - q(D)| \geq 18\varepsilon\right] \leq \delta/\varepsilon.$$

G.2. Template: Privacy $\implies$ Boosting Generalizes

Referring to the boosting schema in Section 3, we now outline how to use the generalization properties of differential privacy to obtain a PAC learner from a private weak learner, via boosting. Recall that the fundamental boosting theorem is a *round bound:* after a certain number of rounds, boosting produces a collection of weak hypotheses that can be aggregated to predict the *training data* well.

Theorem 32 (Template for a Boosting Theorem) *Fix `NxM`. For any weak learning algorithm `WkL` with advantage γ, running `Boost` using these concrete subroutines terminates in at most $T(\gamma, \alpha, \beta)$ steps and outputs (with probability at least $1 - \beta$) a hypothesis H such that:*

$$\Pr_{(x,y)\sim S}[H(x) \neq y] \leq \alpha$$

We can capture the training error of a learning algorithm using a statistical query. For any hypothesis H, define $\mathrm{err}_H(x, y) = 1$ if $H(x) \neq y$, and $\mathrm{err}_H(x, y) = 0$ otherwise.

Denoting by D the target distribution, PAC learning demands a hypothesis such that $\mathrm{err}_H(D) \leq \alpha$, with high probability. If the boosting process is differentially private, the generalization properties of differential privacy ensure that $\mathrm{err}_H(S)$ and $\mathrm{err}_H(D)$ are very close with high probability. Thus, boosting private weak learners can enforce low test error. We elaborate below.

Theorem 33 (Abstract Generalization) *Let `WkL` be a (ρ, ζ)-zCDP weak learner with advantage γ. Suppose `NxM` is ζ-slick and enjoys round-bound T with error α and failure probability β. Denote by $\mathcal{M}'$ the algorithm `Boost` run using `WkL` and `NxM`. Let $\varepsilon = O(\sqrt{\rho T}\log(1/\delta))$ and suppose $n \geq \Omega(\log(\varepsilon/\delta)/\varepsilon^2)$. Then, with probability at least $1 - \beta - \delta/\varepsilon$ over $S \sim_{iid} D^n$ and the internal randomness of $\mathcal{M}'$, the hypothesis output by $\mathcal{M}'$* generalizes *to D:*

$$\Pr_{(x,y)\sim D}[H(x) \neq y] \leq \alpha.$$

Proof [sketch] By the round-bound and inspection of Algorithm L.2, $\mathcal{M}'$ simply composes T calls to `Iter` and post-processes. So by zCDP composition (Lemma 14) we know that $\mathcal{M}'$ is ρT-zCDP. This can be converted to an (ε, δ)-DP guarantee on $\mathcal{M}'$ for any $\delta > 0$ (Lemma 15).

For the sake of analysis, define a new mechanism $\mathcal{M}$ that runs $\mathcal{M}'(S)$ to obtain H and then outputs the statistical query err_H. This is just post-processing, so $\mathcal{M}$ is also (ε, δ)-DP. Thus, given enough samples, the conversion of privacy into generalization for statistical queries applies to $\mathcal{M}$:

$$\Pr_{S\sim D^n}[|\operatorname{err}_H(S) - \operatorname{err}_H(D)| \geq 18\varepsilon] \leq \delta/\varepsilon \text{ (Theorem 31)}\ .$$

By the guarantee of the round-bound, $\operatorname{err}_H(S) \leq \alpha$ with probability at least $1-\beta$. Therefore,

$$\Pr_{S\sim D^n}\left[\Pr_{(x,y)\sim D}[H(x) \neq y] \leq \alpha + 18\varepsilon\right] \leq \delta/\varepsilon + \beta.$$

■

Observe that we require privacy both for privacy's sake and for the generalization theorem. Whichever requirement is more stringent will dominate the sample complexity of any algorithm so constructed.

G.3. Template: Privacy $\implies$ Noise-Tolerant Generalization

Suppose now that there is some kind of interference between our learning algorithm and the training examples. For example, this could be modeled by random classification noise with rate η (Angluin and Laird, 1987). This altered setting violates the preconditions of the DP to generalization transfer. A noised sample is *not* drawn i.i.d. from D and so the differential privacy of $\mathcal{M}$ is not sufficient to guarantee generalization of the "low training error" query err_H as defined above.

To get around this issue, we fold a noise model into the generalization-analysis mechanism. Define an alternative *noised* mechanism $\mathcal{M}_\eta$ (Algorithm 5) atop any $\mathcal{M}$ that outputs a "test error" query, and apply "DP to Generalization" on $\mathcal{M}_\eta$ instead. Suppose that $\mathcal{M}$ is differentially private, and the underlying learning algorithm $\mathcal{A}$ run by $\mathcal{M}$ tolerates noise at rate η. Then, if $\mathcal{M}_\eta$ is DP, we can generalize the noise-tolerance of $\mathcal{A}$.

Algorithm 5 $\mathcal{M}_\eta$ for RCN. Input: $S \in X^n, S \sim D^n$

```
  $\forall i \in [n]\ \mathsf{F}_i \leftarrow 1$
  $\forall i \in [n]\ \mathsf{F}_i \leftarrow -1$ with probability $\eta$
  $\tilde{y}_i \leftarrow \mathsf{F}_i \times y_i$
  $\tilde{S} \leftarrow \{(x_i, y_i)\}$
  $\operatorname{err}_H \leftarrow \mathcal{M}(\tilde{S})$
  return $\operatorname{err}_H$
```

At least for random classification noise, $\mathcal{M}_\eta$ does indeed maintain privacy. Observe that for a fixed noise vector $\boldsymbol{N}$, $\mathcal{M}_\eta$ run on neighboring data sets S and S' will run $\mathcal{M}$ with neighboring datasets $\tilde{S}$ and $\tilde{S}'$, and therefore the output distributions over queries will have bounded distance. Since the noise is determined independent of the sample, this means that $\mathcal{M}_\eta$ inherits the differential privacy of $\mathcal{M}$, and therefore satisfies the conditions of Theorem 31. So the resulting learner still generalizes.

This trick could handle much harsher noise models. For instance, each example selected for noise could be arbitrarily corrupted instead of given a flipped label. But we seem unable to capture fully malicious noise: an adversary viewing the whole sample could compromise privacy and so generalization. Thus, the "effective noise model" implicit above seems to distinguish between adversaries who have a global versus local view of the "clean" sample. This seems a natural division; we hope that future work will explore the expressiveness of this noise model.

Appendix H. Privacy-Only Noise-Tolerant Sample Bound for Large-Margin Halfspaces

We state and prove the formal version of Theorem 2.

Theorem 34 (Learning Halfspaces Under Random Label Noise) *The* `HS-StL` *procedure is a* (ε, δ)*-Differentially Private* (α, β)*-strong PAC learner for* τ*-margin halfspaces tolerating random label noise at rate* $\eta = O(\alpha\tau)$ *with sample complexity*

$$n = \tilde{\Omega}\left(\underbrace{\frac{1}{\varepsilon\alpha\tau^2}}_{\substack{\textit{privacy} \\ \textit{(Claim 1)}}} + \underbrace{\frac{1}{\alpha^2\tau^2}}_{\textit{accuracy}} + \underbrace{\frac{1}{\varepsilon^2} + \frac{1}{\alpha^2}}_{\substack{\textit{generalization} \\ \textit{(Claim 4)}}} \right)$$

Proof Denote by ε_T and δ_T the parameters of approximate differential privacy at the final round T of `HS-StL`, and by the H the output hypothesis of `HS-StL`. We proceed as follows.

1. Given enough samples, `HS-StL` is differentially private. (Claim 1)
2. Random Label Noise at rate $\eta = O(\alpha\tau)$ will (w.h.p.) not ruin the sample. (Claim 2)
3. The Gaussian Mechanism will (w.h.p.) not ruin the weak learner. (Claim 3)
4. Given enough samples, training error is (w.h.p.) close to test error. (Claim 4)
5. Given enough samples, `HS-StL` (w.h.p.) builds a hypothesis with low test error.

For the remainder of this proof, fix the settings of all parameters as depicted in `HS-StL` (Algorithm 5.3). We reproduce them here:

$$\kappa \leftarrow \alpha/4 \tag{1}$$

$$\sigma \leftarrow \tau/8c\sqrt{\log\left(\frac{3072\log(1/\kappa)}{\beta\tau^2}\right)} \tag{2}$$

Claim 1 (Enough Samples $\implies$ `HS-StL` is Differentially Private) *For every* $\delta_T > 0$*, we have:*

$$n > \tilde{O}\left(\frac{1}{\varepsilon_T\alpha\tau^2}\right) \implies \texttt{HS-StL} \textit{ is } (\varepsilon_T, \delta_T)\textit{-DP}$$

Proof By the privacy bound for $\widehat{\mathrm{WL}}$ (Theorem 22), we know that a single iteration of `Boost` is ρ-zCDP for $\rho = \frac{8}{(\kappa n\sigma)^2}$. Then, `Boost` runs for $T = \frac{1024\log(1/\kappa)}{\tau^2}$ rounds. So, by tight composition for zCDP (Lemma 14), `HS-StL` is ρ_T-zCDP where:

$$\rho_T = O\left(\frac{\log(1/\kappa)}{(\kappa n\sigma\tau)^2}\right)$$

Now we convert from zero-concentrated to approximate differential privacy, via Lemma 15: if $\varepsilon_T < 3\sqrt{\rho_T\log(1/\delta_T)}$, then `HS-StL` is $(\varepsilon_T, \delta_T)$-DP for all $\delta_T > 0$. We re-arrange to bound n.

$$\rho_T < O\left(\frac{\varepsilon_T^2}{\log(1/\delta_T)}\right)$$

Unpacking ρ_T we get:

$$\frac{\log(1/\kappa)}{(\kappa n \sigma \tau)^2} < O\left(\frac{\varepsilon_T^2}{\log(1/\delta_T)}\right)$$

This will hold so long as:

$$n > \Omega\left(\frac{\sqrt{\log(1/\kappa)\log(1/\delta_T)}}{\kappa\sigma\tau\varepsilon_T}\right)$$

Substituting the settings of σ and κ from `HS-StL`, we obtain:

$$n > \Omega\left(\frac{\sqrt{\log(1/\alpha)\log(1/\delta_T)\log(\log(1/\alpha)/\beta\tau^2)}}{\varepsilon_T \alpha \tau^2}\right)$$

■

We next consider the two events that could destroy good training error.

Too Many Corrupted Samples Noise could corrupt so many samples that the weak learner fails. Under an approprite noise rate, this is unlikely. We denote this event by BN (for "bad noise").

Gaussian Mechanism Destroys Utility The Gaussian noise injected to ensure privacy could destroy utility for a round of boosting. We denote this event by BG (for "bad Gaussian").

Both events are unlikely, under the settings of `HS-StL`.

Claim 2 (Hopelessly Corrupted Samples are Unlikely) *Let $\mathsf{F}_1, \ldots, \mathsf{F}_n$ indicate the event "label i was flipped by noise," and denote by $\mathsf{F} = \sum_{i=1}^n \mathsf{F}_i$ the number of such corrupted examples. Under the settings of `HS-StL` and noise rate $\eta = \alpha\tau/32$, we have:*

$$n > \frac{96\ln(4/\beta)}{\alpha\tau} \implies \Pr[\mathsf{BN}] = \Pr[\mathsf{F} > \kappa n] \leq \beta/4$$

Proof At noise rate η, we have $\mathbb{E}[\mathsf{F}] = n\eta$. From the definitions and Theorem 22,

$$\Pr[\mathsf{BN}] = \Pr\left[\mathsf{F} \geq \frac{\alpha\tau}{16}n\right]$$

We apply the following simple Chernoff bound: $\forall \delta \geq 1$

$$\Pr[\mathsf{F} \geq (1+\delta)\mathbb{E}[\mathsf{F}]] \leq \exp(-\mathbb{E}[\mathsf{F}]\,\delta/3)$$

Substituting with $\delta = 1$:

$$\Pr[\mathsf{F} \geq 2\eta n] \leq \exp(-(\eta n)/3)$$

Noise rate $\eta = \alpha\tau/32$ gives the appropriate event above:

$$\Pr\left[\mathsf{F} \geq 2\eta n\right] = \Pr\left[\mathsf{F} \geq \frac{\alpha\tau}{16} n\right] \leq \exp(-(\alpha\tau n)/96)$$

Constraining the above probability to less than β/k_1 for any constant $k_1 > 1$ we solve to obtain:

$$n > \frac{96\ln(k_1/\beta)}{\alpha\tau}$$

■

Claim 3 (Bad Gaussians are Unlikely) *Let BG_i indicate the event that the ith call to the weak learner fails to have advantage at least $\tau/8$. Under the settings of* `HS-StL`*:*

$$\Pr[\mathsf{BG}] = \Pr[\exists i\ \mathsf{BG}_i] \leq \beta/2$$

Proof Our setting of σ simplifies the advantage of $\widehat{\mathrm{WL}}$ to $\gamma(\tau,\sigma) = \tau/8$ with all but probability $\xi = \frac{\beta\tau^2}{1024\log(1/\kappa)}$. Then, by the round bound for `LB-NxM` (Theorem 20), `Boost` will terminate after $T = \frac{8\log(1/\kappa)}{\gamma^2} = \frac{512\log(1/\kappa)}{\tau^2}$ rounds, and so we have that with probability $(1-\xi)^T \geq 1 - \frac{\beta}{2}$ every hypothesis output by $\widehat{\mathrm{WL}}$ satisfies the advantage bound.

■

To enforce generalization, we capture both training and test error for any hypothesis H with a statistical query that indicates misclassification. Evaluated over the sample it is the training error of H, and evaluated over the population it is the test error of H.

$$\mathrm{err}_H(x,y) \mapsto \begin{cases} 1 & \text{if } yH(x) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

Claim 4 (Enough Samples $\implies$ Good Generalization) *If $0 < \varepsilon_T < 1/3$ and $0 < \delta_T < \varepsilon_T/4$*

$$n \geq \tilde{\Omega}\left(\frac{1}{\varepsilon_T^2}\right) \implies \Pr_{S\sim D^n}\left[\mathrm{err}_H(D) \geq \mathrm{err}_H(S) + 18\varepsilon_T\right] \leq \delta_T/\varepsilon_T$$

Proof Define (for analysis only) a procedure `HS-StL-test`, which outputs the "error" statistical query. That is, letting $H = \texttt{HS-StL}(S)$ where $S \sim D^n$, `HS-StL-test` prints err_H. Thus, `HS-StL-test` is a mechanism for selecting a statistical query.

Because `HS-StL-test` is simple post-processing, it inherits the privacy of `HS-StL`. Since we select err_H privately, it will (by Theorem 31) be "similar" on the sample S (training error) and the population D (test error). Ignoring over-estimates of test error and observing the sample bounds of Theorem 31, this gives the claim.

■

Claim 5 (Low Training Error is Likely) *Given $\neg\mathsf{GB}$ and $\neg\mathsf{BN}$, we have $\mathrm{err}_H(S) \leq \alpha/2$.*

Proof Let $\tilde{S}$ denote the noised sample, and let S_C and S_D be the "clean" and "dirty" subsets of examples, respectively.

Given $\neg\mathsf{GB}$ and $\neg\mathsf{BN}$, the weak learning assumption holds on every round. So, by Theorem 20, the boosting algorithm will attain low training error $\mathrm{err}_H(\tilde{S}) = \kappa$. This is not yet enough to imply low test error, because $\tilde{S} \not\sim_{iid} D$. So we bound $\mathrm{err}_H(S)$ using $\mathrm{err}_H(\tilde{S})$. Suppose that the noise affects training in the worst possible way: H fits *every* flipped label, so H gets *every* example in S_D wrong. Decompose and bound $\mathrm{err}_H(S)$ as follows:

$$\begin{aligned}
\mathrm{err}_H(S) &= \sum_{(x,y)\in S} \mathrm{err}_H(x,y) \\
&= \sum_{(x,y)\in S_C} \mathrm{err}_H(x,y) + \sum_{(x,y)\in S_D} \mathrm{err}_H(x,y) \\
&\leq \kappa + |S_D| && \text{Boosting Theorem, worst case fit} \\
&\leq \kappa + \frac{\alpha\tau}{16} && \neg\mathsf{BN}
\end{aligned}$$

Because the sample is from the unit ball, we have $\tau \in (0,1)$. Therefore, it is always the case that $\frac{\alpha\tau}{16} < \frac{\alpha}{4}$. So $\mathrm{err}_H(S) \leq \kappa + \alpha/4 \leq \alpha/2$, concluding proof of the above claim. ■

It remains only to select ε_T and δ_T so that the claims above may be combined to conclude low test error with high probability. Recall that our objective is sufficient privacy to simultaneously:

1. Ensure that `HS-StL` is (ε, δ)-DP
2. Apply DP to generalization transfer (Theorem 31) for good test error.

Both these objectives impose constraints on ε_T and δ_T. The requirement that the algorithm is desired to be (ε, δ)-DP in particular forces ε_T to be smaller than ε and δ_T to be smaller than δ. The transfer theorem is slightly more subtle; to PAC-learn, we require:

$$\Pr_{S\sim D^n}[\mathrm{err}_H(D) \geq \alpha] \leq \beta$$

While Claim 4 gives us that

$$\Pr_{S\sim D^n}[\mathrm{err}_H(D) \geq \mathrm{err}_H(S) + 18\varepsilon_T] \leq \delta_T/\varepsilon_T$$

So, accounting for both privacy and accuracy, we need $\varepsilon_T < \phi = \min(\varepsilon, \alpha/36)$. We can select any $\delta_T < \min(\delta,\ \phi\beta/4)$ to ensure that $\delta_T/\varepsilon_T < \beta/2$. By substituting the different realizations of these 'min' operations into Claims 1 and 4 we obtain the sample bound.

Finally, observe that with these settings we can union bound the probability of BN and BG and the event that generalization fails with β, as required for PAC learning. But it follows from the claims above that if $\neg\mathsf{BN}$ and $\neg\mathsf{BG}$ and a good transfer all occur, then the ouput hypothesis H has test error less than α, concluding the argument.

■

Appendix I. Smooth Boosting via Games

Here, we prove our round-bound and final margin guarantee for `LazyBregBoost`. The proof is a reduction to approximately solving two-player zero-sum games. We introduce the basic elements of game theory, then outline and excute the reduction. Overall, we recall and prove Theorem 20:

Theorem 20 (Lazy Bregman Round-Bound) *Suppose we run* `Boost` *with* `LB-NxM`$(\kappa, \gamma/4)$ *on a sample* $S \subset \mathcal{X} \times \{\pm 1\}$ *using any real-valued weak learner with advantage* γ *for* $T \geq \frac{16 \log(1/\kappa)}{\gamma^2}$ *rounds. Let* $H : \mathcal{X} \to [-1, 1]$ *denote the final, aggregated hypothesis. The process has:*

Good Margin: *H mostly agrees with the labels of S.*

$$\Pr_{(x,y)\sim S} [yH(x) \leq \gamma] \leq \kappa$$

Smoothness: *Every distribution $\hat{\mu}_t$ supplied to the weak learner has $\hat{\mu}_t(i) \leq \frac{1}{\kappa n}$ $\forall i$*

I.1. Two-Player Zero-Sum Games

A two player game can be described by a matrix, where the *rows* are indexed by "row player" strategies $\mathcal{P}$, the *columns* are indexed by "column player" strategies $\mathcal{Q}$, and each entry (i, j) of the matrix is the *loss* suffered by the row player when row strategy $i \in \mathcal{P}$ is played against column strategy $j \in \mathcal{Q}$. Such a game is *zero-sum* when the colum player is given as a reward the row player's loss. Accordingly, the row player should minimize and the column player should maximize.

A single column or row is called a *pure strategy.* To model Boosting, we imagine players who can randomize their actions. So the fundamental objects are *mixed strategies:* distributions P over the rows and Q over the columns. Playing "according to" a mixed strategy means sampling from the distribution over pure strategies and playing the result. When two mixed strategies are played against each other repeatedly, we can compute the *expected loss* of P vs. Q playing the game M:

$$M(P, Q) = \underbrace{\sum_{i,j \in \mathcal{P} \times \mathcal{Q}} P(i) M(i, j) Q(j)}_{\text{(i)}} = \underbrace{\sum_{j \in \mathcal{Q}} M(P, j) Q(j)}_{\text{(ii)}} = \underbrace{\sum_{i \in \mathcal{P}} P(i) M(i, Q)}_{\text{(iii)}} \tag{3}$$

"Iterated play" pits the row player against an arbitrary environment represented by the column player. At each round, both the row player and column player choose strategies P_t and Q_t respectively. The expected loss of playing Q_t against each *pure* row strategy is revealed to the row player. Then, the row player suffers the *expected loss* of P_t vs. Q_t. This set-up is depicted by Algorithm 6. Good row player strategies have *provably bounded regret* — they do not suffer much more loss than the best possible *fixed* row player strategy in hindsight during iterated play.

Here, we reduce boosting to the "Lazy Dense Multiplicative Updates" row player strategy (Algorithm 7) which enjoys bounded regret (Lemma 35, proved in Appendix J for completeness) and two other helpful properties:

Simple State: The only state is all previous loss vectors and step count so far; this enables privacy.

Single Projection: It Bregman-projects just once per round; this enforces slickness.

Algorithm 6 Iterated Play

Input: T the number of rounds to play for

Output: Total expected row player cost incurred

```
for $t = 1$ to $T$ do
    $P_t \leftarrow$ Row player choice of mixed strategies, seeing $\ell_1, \dots, \ell_{t-1}$
    $Q_t \leftarrow$ Column player choice of mixed strategies, seeing $P_t$
    $\ell_t(i) \leftarrow M(i, Q_t)\ \forall i$   /* Reveal loss on each pure row strategy */
    $C \leftarrow C + M(P_t, Q_t)$   /* Accumulate total loss */
end for
```

Algorithm 7 Lazy Dense Update Strategy (LDU)

Input: $\mathcal{P}$, a set of pure row-player strategies, learning rate λ, losses $\ell_1, \dots, \ell_T$

Output: A measure over $\mathcal{P}$

```
for $i \in \mathcal{P}$ do
    $\mu_1(i) \leftarrow \kappa$
end for
for $i \in \mathcal{P}$ do
    $\tilde{\mu}_{T+1}(i) \leftarrow e^{-\lambda \sum_{t=1}^{T} \ell_t(i)} \mu_1(i)$
end for
$\mu_{T+1} \leftarrow \Pi_\Gamma \tilde{\mu}_T$
```

Lemma 35 (Lazy Dense Updates Regret Bound) *Let Γ be the set of κ-dense measures. Set $\mu_1(i) = \kappa$ for every i. Then for all $\mu \in \Gamma$ we have the following regret bound.*

$$\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_t, Q_t) \leq \frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \lambda + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda\kappa|\mathcal{P}|T}$$

I.2. Reducing Boosting to a Game

We present a reduction from boosting real-valued weak learners to approximately solving iterated games, following Freund and Schapire (1996). To prove the necessary round-bound (Theorem 20), we do the following:

1. **Create a Game.** The meaning of "advantage" given by a Weak Learning assumption (Definition 11) naturally induces a zero-sum game where pure row strategies are points in the sample S and pure column strategies are hypotheses in $\mathcal{H}$. The Booster will play mixed row strategies by weighting the sample and the weak learner will play pure column strategies by returning a single hypothesis at each round.

2. **Weak Learning $\implies$ Booster Loss Lower-Bound.** Weak Learners have some advantage in predicting with respect to *any* distribution on the sample. Thus, the particular sequence of distributions played by *any* Booster must incur at least some loss.

3. **Imagine Pythia, a Prophetic Booster.** Given perfect foreknowledge of how the weak learner will play, what is the best Booster strategy? Create a "prescient" Boosting strategy $P^\star$ which

concentrates measure on the "worst" examples $(x, y) \in S$ for the *final* hypothesis H at each round.

4. **How Well Does Pythia Play?** Upper-bound the total loss suffered by a Booster playing $P^\star$ each round.

5. **Solve for T:** Recall that we want the Booster to *lose*. That is, we want an accurate ensemble of hypotheses. Combining the upper and lower bounds above with the regret bound, we solve for a number of rounds to "play" (Boost) for such that the size of the set of "worst" examples shrinks to a tolerable fraction of the sample. This gives the round-bound (Theorem 20).

I.2.1. CREATE A GAME

Our game is a continuous variant of the Boolean "mistake matrix" (Freund and Schapire, 1996). Let $\mathcal{H} \subseteq \{B_2(1) \to [-1, 1]\}$ be a set of bounded $\mathbb{R}$-valued hypothesis functions on the unit ball. These functions will be the pure column player strategies. Now let $S = (x_1, y_1), \ldots, (x_n, y_n)$ be a set of points where $y_i \in \{\pm 1\}$ and $x_i \in B_2(1)$. These points will be the pure row player strategies. Having chosen the strategies, all that remains is to define the entries of a matrix. To cohere with the definition of weak learning for real-valued functions, we define the following game of *soft punishments*:

$$M_S^{\mathcal{H}} := M_S^{\mathcal{H}}(i, h) = 1 - \frac{1}{2}|h(x_i) - y_i|$$

Notice the quantification here: we can define the soft punishment game for any sample and any set of hypotheses. We will omit $\mathcal{H}$ and S when fixed by context. This is a simple generalization of the game from Freund and Schapire (1996) which assigns positive punishment 1 to correct responses, and 0 to incorrect responses. Since we work with real-valued instead of Boolean predictors, we alter the game to scale row player loss by how "confident" the hypothesis is on an input point.

I.2.2. WEAK LEARNING $\implies$ BOOSTER LOSS LOWER-BOUND.

Mixed strategies for the booster (row player) are just distributions over the sample. So the accuracy assumption on the weak learner `WkL`, which guarantees advantage on *every* distribution, induces a lower bound on the total loss suffered by *any* booster playing against `WkL`. Recall that losses are measured with respect to *distributions* over strategies, not *measures*. So, below we normalize any measure to a distribution "just before" calculating expected loss

Lemma 36 (Utility of Weak Learning) *For any sequence of T booster mixed strategies $(\mu_1, \ldots, \mu_T)$, suppose the sequence of column point strategies $h_1, \ldots, h_T$ is produced by a weak learner that has advantage γ. Then:*

$$\sum_{t=1}^{T} M(\hat{\mu}_t, h_t) \geq {}^{T}\!/_{2} + T\gamma$$

Proof

$$\begin{aligned}
\sum_{t=1}^{T} M(\hat{\mu}_t, h_t) &= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\hat{\mu}_t} [M(i, h_t)] && \text{unroll def 3 } (iii) \\
&= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\hat{\mu}_t} [1 - {}^1\!/\!_2|h_t(x_i) - y_i|] && \text{re-arrange ``advantage''} \\
&= \sum_{t=1}^{T} {}^1\!/\!_2 + \mathop{\mathbb{E}}_{i\sim\hat{\mu}_t} [{}^1\!/\!_2 - {}^1\!/\!_2|h_t(x_i) - y_i|] && \text{linearity of } \mathbb{E} \\
&= {}^T\!/\!_2 + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\hat{\mu}_t} [{}^1\!/\!_2 h_t(x_i) y_i] && \text{distributing summations} \\
&\geq {}^T\!/\!_2 + T\gamma && \text{by Weak Learning Assumption}
\end{aligned}$$

■

I.2.3. IMAGINE PYTHIA, A PROPHETIC BOOSTER.

How should a booster play if she knows the future? Suppose Pythia knows exactly which hypotheses $h_1, \ldots, h_T$ the weak learner will play, but is restricted to playing the same fixed κ-dense strategy for all T rounds. Intuitively, she should assign as much mass as possible to points of S where the combined hypothesis $H = (1/T)\sum_{t\in[T]} h_t$ is incorrect, and then assign remaining mass to points where H is correct but uncertain. We refer to this collection of points as B, the set of "bad" points for $h_1, \ldots, h_T$. We formalize this strategy as Algorithm 8.

Algorithm 8 Pythia

```
Input: $S$ a sample with $|S| = n$; $H$ a combined hypothesis; $\kappa$ a target density
Output: Distribution $P^\star$ over $[n]$; Minimum margin $\theta_T$
  $B \leftarrow \{i \in [1, n] \mid y_i H(x_i) < 0\}$    /⋆ Place all mistakes in $B$ ⋆/
  Sort $[1, n] \setminus B$ by margin of $H$ on each point
  $\theta_T \leftarrow 0$
  while $|B| < \kappa n$ do
    Add minimum margin element $i$ of $[1, n] \setminus B$ to $B$
    Update $\theta_T$ to margin of $H$ on $(x_i, y_i)$
  end while
  $P^\star \leftarrow$ the uniform distribution over $B$
  Output $P^\star, \theta_T$
```

The prophetic booster Pythia plays the uniform measure on a set B of "bad" points selected by Algorithm 8, normalized to a distribution. That is:

$$P^\star(i) = \begin{cases} 1/|B| & \text{if } i \in B \\ 0 & \text{otherwise} \end{cases}$$

It is important to observe that if i is outside of the "bad set" B, we know H has "good" margin on (x_i, y_i). To quantify this, observe that for all $i \in B$, H has margin at most θ_T on (x_i, y_i).

Proposition 37 (Bad Margin in Bad Set) *For every $i \in B$, we know $\sum_{t=1}^{T} y_i h(x_i) \leq T\theta_T$*

Proof

$$\begin{aligned} i \in B \implies y_i H(x_i) &\leq \theta_T && \text{inspection of Pythia, above} \\ \frac{y_i}{T}\sum_{t=1}^{T} h_t(x_i) &\leq \theta_T && \text{unroll } H \\ \sum_{t=1}^{T} y_i h_t(x_i) &\leq T\theta_T && \text{re-arrange} \end{aligned}$$

■

I.2.4. How Well Does Pythia Play?

Here, we calculate the utility of foresight — an upper-bound on the loss of $P^\star$. Suppose H is the terminal hypothesis produced by the boosting algorithm. We substitute $P^\star$ into the definition of expected loss for $M_S^{\mathcal{H}}$ (soft punishments) and relate the margin on the bad set for H to the cumulative loss of H, giving the following lemma.

Lemma 38 (Excellence of Pythia) *Let S be a sample, $(h_1, \ldots, h_T) \in \mathcal{H}^T$ a sequence of hypotheses, $H = (1/T)\sum_{i=1}^{T} h_i$, and $\kappa \in [0, 1/2]$ a density parameter. Let $P^\star, \theta_H = \mathrm{Pythia}(S, H, \kappa)$. Then:*

$$\sum_{t=1}^{T} M(P^\star, h_t) \leq (T/2) + (T\theta_H)/2$$

We require a simple fact about advantages. Since $h(x) \in [-1, +1]$ and $y \in \{\pm 1\}$, we know:

$$\begin{gathered} yh(x) = 1 - |h(x) - y| \\ \implies (1/2)(yh(x)) = (1/2) - (1/2)|h(x) - y| \end{gathered}$$

The entries of the soft punishments matrix can also be re-written by formatting advantage as above. For $i \in [1, n]$ and $h \in \mathcal{H}$ we have:

$$\mathrm{M}(i, h) = (1/2) + (1/2)(y_i h(x_i)) \tag{4}$$

Proof We manipulate the total regret of $P^\star$ towards getting an upper-bound in terms of the minimum margin of H and number of rounds played.

$$\begin{aligned}
\sum_{t=1}^{T} M(P^\star, h_t) &= \sum_{t=1}^{T}\sum_{i=1}^{n} P^\star(i)\cdot \mathrm{M}(i,h_t) && \text{Part (iii) of Expected Loss (Definition 3)}\\
&= \sum_{t=1}^{T}\sum_{i\in B} P^\star(i)\cdot \mathrm{M}(i,h_t) && \text{Restrict sum — } P^\star(i)=0 \text{ outside } B\\
&= \frac{1}{|B|}\sum_{t=1}^{T}\sum_{i\in B} \mathrm{M}(i,h_t) && \text{Factor out } P^\star(i) \text{ — constant by definition}\\
&= \frac{1}{|B|}\sum_{i\in B}\sum_{t=1}^{T} ((1/2)+(1/2)h_t(x_i)y_i) && \text{Equation 4 about } M_S^{\mathcal{H}} \text{ entries}\\
&= \frac{1}{|B|}\left((|B|T)/2 + (1/2)\sum_{i\in B}\sum_{t=1}^{T} h_t(x_i)y_i\right) && \text{Algebra}\\
&\le \frac{1}{|B|}\left((|B|T)/2 + (1/2)\sum_{i\in B} T\theta\right) && \text{Bad margin in } B \text{ (Proposition 37)}\\
&= (T/2) + (T\theta)/2 && \text{Evaluate \& re-arrange}
\end{aligned}$$

■

I.2.5. SOLVE FOR T.

We now have an upper bound on the loss incurred by a prescient booster, and a lower bound on the loss to *any* booster under the weak learning assumption. This allows us to “sandwich” the performance of boosting according to the lazy dense updates (LDU, Algorithm 7) strategy between these two extremes, because LDU has good performance relative to *any* fixed strategy (Lemma 35, proved in Appendix J). This sandwich gives a relationship between the number of rounds T and the margin of the final hypothesis, which we now solve for the number of rounds necessary to boost using LDU to obtain a “good” margin on “many” samples.

Lemma 39 *Let S be a sample of size n, let μ_t be the measure produced at round t by $\texttt{NxM}(S, H_{t-1})$ playing the Lazy Dense Update Strategy of Algorithm 7, and let h_t be the hypothesis output by $\widehat{\texttt{WkL}}(S, \hat{\mu}_{t-1}, \sigma)$ at round t. Then after $T \ge \frac{16\log(1/\kappa)}{\gamma^2}$ rounds of $\texttt{Iter}$, the hypothesis $H_T(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$ has margin at least γ on all but κn many samples.*

Proof Denote by $\mathcal{U}(B)$ the uniform measure (all $x \in B$ assigned a weight of 1) on the bad set B discovered by Pythia. Combining the regret bound comparing LDU to fixed Pythia with the lower bound on loss that comes from the weak learner assumption, we have, overall:

$$\frac{T}{2} + T\gamma \underbrace{\le}_{\substack{\text{Weak Learning}\\ \text{(Lemma 36)}}} \sum_{t=1}^{T} \mathrm{M}(\hat{\mu}_t, h_t) \underbrace{\le}_{\substack{\text{Regret Bound}\\ \text{(Lemma 35)}}} \sum_{t=1}^{T} \mathrm{M}(P^\star, h_t) + \lambda T + \frac{\mathrm{KL}(\mathcal{U}(B) \parallel \mu_1)}{\kappa n \lambda}$$

Apply Lemma 38, replacing prescient Boosting play with the upper bound on loss we obtained:

$$T\gamma \leq \sum_{t=1}^{T} \mathrm{M}(\hat{\mu}_t, h_t) \leq \frac{T\theta_H}{2} + \lambda T + \frac{\mathrm{KL}(\mathcal{U}(B) \parallel \mu_1)}{\kappa n \lambda}$$

Let's compute the necessary KL-divergence, recalling that $|\mathcal{U}(B)| = |\mu_1| = \kappa n$:

$$\begin{aligned}\mathrm{KL}(\mathcal{U}(B) \parallel \mu_1) &= \sum_{x \in B} \mathcal{U}(B)(x) \log\left(\frac{\mathcal{U}(B)(x)}{\mu_1(x)}\right) - |\mathcal{U}(B)| + |\mu_1| \\ &= \sum_{x \in B} \log\left(\frac{1}{\kappa}\right) \\ &= \kappa n \log\left(\frac{1}{\kappa}\right)\end{aligned}$$

Substituting into the above and dividing through by T, we have:

$$\gamma \leq \frac{\theta_H}{2} + \lambda + \frac{\log(1/\kappa)}{T\lambda}$$

Which, setting

$$T = \frac{16 \log\left({}^{1}\!/\!{}_{\kappa}\right)}{\gamma^2} \text{ and } \lambda = \gamma/4$$

implies

$$\theta_H \geq \gamma.$$

This is the margin bound we claimed, for every $(x, y) \notin B$. ■

Appendix J. Bounded Regret for Lazily Projected Updates

A "lazy" multiplicative weights strategy that, at each round, projects only *once* into the space of dense measures is presented below. Here, we prove that this strategy (Algorithm 9) has bounded regret relative to any fixed dense strategy.

Algorithm 9 Lazy Dense Update Process

```
Input: $\mathcal{P}$, a set of pure row-player strategies, learning rate $\lambda$, losses $\ell_1, \ldots, \ell_T$
Output: A measure over $P$
  for $x \in \mathcal{P}$ do
    $\mu_1(x) \leftarrow \kappa$
  end for
  for $x \in \mathcal{P}$ do
    $\tilde{\mu}_{T+1}(x) \leftarrow e^{-\lambda \sum_{t=1}^{T} \ell_t(x)} \mu_1(x)$
  end for
  $\mu_{T+1} \leftarrow \Pi_\Gamma \tilde{\mu}_T$
```

To analyze the regret of a row player playing the strategy of Algorithm 9, we will need the following definition.

Definition 40 (Strong convexity) *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is α-strongly convex on $X \subset \mathbb{R}^n$ with respect to the ℓ_p norm if for all $x, y \in X$*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha \|x - y\|_p^2.$$

If f is twice differentiable, then f is α-strongly convex on X with respect to the ℓ_p norm if for all $x \in X$, $y \in \mathbb{R}^n$

$$y^T(\nabla^2 f(x))y \geq \alpha \|y\|_p^2.$$

We now proceed to analyze Algorithm 9 by arguments closely following those found in Chapter 2 of Rakhlin's notes on online learning (Rakhlin, 2009). First, we show this update rule minimizes a sequential regularized loss over *all* dense measures.

Lemma 41 *Let $M(\hat{\mu}, Q_t) = \mathbb{E}_{i \sim \hat{\mu}}[\ell_t(i)]$. Let Γ be the set of κ-dense measures, and let μ_1 be the uniform measure over $\mathcal{P}$ of density κ ($\mu_1(x) = \kappa$ for all $x \in P$). Then for all $T \geq 1$, the measure μ_{T+1} produced by Algorithm 9 satisfies*

$$\mu_{T+1} = \arg\min_{\mu \in \Gamma} \left[\lambda |\mu| \sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \mathrm{KL}(\mu \parallel \mu_1) \right]$$

Proof Suppose towards contradiction that there exists $\mu \in \Gamma$ such that

$$\lambda |\mu| \sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \mathrm{KL}(\mu \parallel \mu_1) < \lambda |\mu_{T+1}| \sum_{t=1}^{T} M(\hat{\mu}_{T+1}, Q_t) + \mathrm{KL}(\mu_{T+1} \parallel \mu_1).$$

Then it must be the case that

$$\begin{aligned}
\mathrm{KL}(\mu_{T+1} \parallel \mu_1) - \mathrm{KL}(\mu \parallel \mu_1) &> \lambda \sum_{t=1}^{T} \langle \mu - \mu_{T+1}, \ell_t \rangle \\
&= \langle \mu - \mu_{T+1}, \sum_{t=1}^{T} \lambda \ell_t \rangle \\
&= \sum_i (\mu(i) - \mu_{T+1}(i)) \log \tfrac{\mu_1(i)}{\tilde{\mu}_{T+1}(i)} \\
&= \sum_i \mu(i) \left(\log \tfrac{\mu(i)}{\tilde{\mu}_{T+1}} - \log \tfrac{\mu(i)}{\mu_1(i)} \right) - \sum_i \mu_{T+1}(i) \left(\log \tfrac{\mu_{T+1}(i)}{\tilde{\mu}_{T+1}(i)} - \log \tfrac{\mu_{T+1}(i)}{\mu_1(i)} \right) \\
&= \mathrm{KL}(\mu \parallel \tilde{\mu}_{T+1}) - \mathrm{KL}(\mu \parallel \mu_1) - \mathrm{KL}(\mu_{T+1} \parallel \tilde{\mu}_{T+1}) + \mathrm{KL}(\mu_{T+1} \parallel \mu_1).
\end{aligned}$$

Under our assumption, then, it must be the case that $0 > \mathrm{KL}(\mu \parallel \tilde{\mu}_{T+1}) - \mathrm{KL}(\mu_{T+1} \parallel \tilde{\mu}_{T+1})$, but μ_{T+1} was defined to be the κ-dense measure that minimized the KL divergence from $\tilde{\mu}_{T+1}$, and so we have a contradiction. ∎

Using Lemma 41, we can now show the following regret bound for a row player that, at each round, "knows" the column strategy Q_t that it will play against that round.

Lemma 42 *Let Γ be the set of κ-dense measures. Let μ_1 be the uniform measure over $\mathcal{P}$ of density κ ($\mu_1(x) = \kappa$ for all $x \in \mathcal{P}$), and let $|\mathcal{P}| = n$. Then for all $\mu \in \Gamma$ we have that*

$$\sum_{t=1}^{T} M(\hat{\mu}_{t+1}, Q_t) \leq \sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n}$$

Proof For $T = 0$, this follows immediately from the definition of μ_1.

Assume now that for any $\mu \in \Gamma$ that

$$\sum_{t=1}^{T-1} M(\hat{\mu}_{t+1}, Q_t) \leq \sum_{t=1}^{T-1} M(\hat{\mu}, Q_t) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n},$$

and in particular

$$\sum_{t=1}^{T-1} M(\hat{\mu}_{t+1}, Q_t) \leq \sum_{t=1}^{T-1} M(\hat{\mu}_{T+1}, Q_t) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n}.$$

It follows that

$$\begin{aligned} \sum_{t=1}^{T} M(\hat{\mu}_{t+1}, Q_t) &= \sum_{t=1}^{T-1} M(\hat{\mu}_{t+1}, Q_t) + M(\hat{\mu}_{T+1}, Q_t) \\ &= \sum_{t=1}^{T-1} M(\hat{\mu}_{t+1}, Q_t) + \sum_{t=1}^{T} M(\hat{\mu}_{T+1}, Q_t) - \sum_{t=1}^{T-1} M(\hat{\mu}_{T+1}, Q_t) \\ &\leq \sum_{t=1}^{T} M(\hat{\mu}_{T+1}, Q_t) + \frac{\mathrm{KL}(\mu_{T+1} \parallel \mu_1)}{\lambda \kappa n} \\ &\leq \sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n}, \end{aligned}$$

for all $\mu \in \Gamma$, since $\mu_{T+1} = \arg\min_{\mu \in \Gamma} [\lambda |\mu| \sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \mathrm{KL}(\mu \parallel \mu_1)]$. ■

To show a regret bound for the lazy dense update rule of Algorithm 9, we now need only relate the lazy-dense row player's loss to the loss of the row player with foresight. To prove this relation, we will need the following lemma showing strong convexity of $R(\mu) = \mathrm{KL}(\mu \parallel \mu_1)$ on the set of κ-dense measures.

Lemma 43 *The function $R(\mu) = \mathrm{KL}(\mu \parallel \mu_1)$ is $(1/\kappa n)$-strongly convex over the set of measures with density no more than κ, with respect to the ℓ_1 norm.*

Proof Let μ be a measure of density $d \leq \kappa$, and let $x = (\frac{1}{\mu(1)}, \dots, \frac{1}{\mu(n)})$. The Hessian of $R(\mu)$ is $\nabla^2 R(\mu) = I_n x$. Therefore, for all $y \in \mathbb{R}^n$,

$$\begin{aligned} y^T \nabla^2 R(\mu) y &= \sum_i \frac{y_i^2}{\mu(i)} \\ &= \frac{1}{|\mu|} \sum_i \mu(i) \sum_i \frac{y_i^2}{\mu(i)} && \text{multiply by 1} \\ &\geq \frac{1}{|\mu|} \left(\sum_i \sqrt{\mu(i)} \frac{y_i}{\sqrt{\mu(i)}} \right)^2 && \text{Cauchy-Schwarz} \\ &= \frac{1}{|\mu|} \|y\|_1^2 \end{aligned}$$

Therefore $R(\mu)$ is strongly convex on the given domain, for the given norm. ■

We can now relate the losses $M(\hat{\mu}_T, Q_t)$ and $M(\hat{\mu}_{T+1}, Q_t)$.

Lemma 44 *Let $R(\mu) = \mathrm{KL}(\mu \parallel \mu_1)$, which is $1/\kappa n$-strongly convex with respect to the ℓ_1 norm on Γ, the set of κ-dense measures. Then for all $T \geq 1$*

$$M(\hat{\mu}_T, Q_t) - M(\hat{\mu}_{T+1}, Q_t) \leq \lambda$$

Proof We first note that $M(\hat{\mu}_T, Q_t) - M(\hat{\mu}_{T+1}, Q_t) = \frac{1}{\kappa n} \langle \mu_T - \mu_{T+1}, \ell_T \rangle$. So it suffices to show that

$$\sum_i (\mu_T(i) - \mu_{T+1}(i)) \ell_T(i) \leq \kappa n \lambda$$

which, because $\ell_T(i) \in [0, 1]$, is implied by $\|\mu_t(i) - \mu_{t+1}(i)\|_1 \leq \kappa n \lambda$.

Our strong convexity assumption on R and an application of Bregman's theorem (Theorem 8) give us that

$$\begin{aligned} \tfrac{1}{\kappa n} \|\mu_T - \mu_{T+1}\|_1^2 &\leq \langle \nabla R(\mu_T) - \nabla R(\mu_{T+1}), \mu_T - \mu_{T+1} \rangle \\ &= \mathrm{KL}(\mu_T \parallel \mu_{T+1}) + \mathrm{KL}(\mu_{T+1} \parallel \mu_T) \\ &\leq \mathrm{KL}(\mu_T \parallel \tilde{\mu}_{T+1}) - \mathrm{KL}(\mu_{T+1} \parallel \tilde{\mu}_{T+1}) + \mathrm{KL}(\mu_{T+1} \parallel \tilde{\mu}_T) - \mathrm{KL}(\mu_T \parallel \tilde{\mu}_T) \\ &= \sum_i \mu_T(i) \left(\log \frac{\mu_T(i)}{\tilde{\mu}_{T+1}(i)} - \log \frac{\mu_T(i)}{\tilde{\mu}_T(i)} \right) - \sum_i \mu_{T+1}(i) \left(\log \frac{\mu_{T+1}(i)}{\tilde{\mu}_{T+1}(i)} - \log \frac{\mu_{T+1}(i)}{\tilde{\mu}_T(i)} \right) \\ &= \sum_i \mu_T(i) \left(\log \frac{\tilde{\mu}_T(i)}{\tilde{\mu}_{T+1}(i)} \right) - \sum_i \mu_{T+1}(i) \left(\log \frac{\tilde{\mu}_T(i)}{\tilde{\mu}_{T+1}(i)} \right) \\ &= \langle \mu_T - \mu_{T+1}, \lambda \ell_{T+1} \rangle \\ &\leq \|\mu_T - \mu_{T+1}\|_1 \|\lambda \ell_{T+1}\|_\infty \end{aligned}$$

Therefore

$$\tfrac{1}{\kappa n} \|\mu_T - \mu_{T+1}\|_1 \leq \lambda \|\ell_{T+1}\|_\infty \leq \lambda$$

and so

$$\|\mu_T - \mu_{T+1}\|_1 \leq \lambda \kappa n.$$

■

We are now ready to prove the regret bound for the lazy dense update process of Algorithm 9, stated earlier in a simplified form as Lemma 35.

Lemma 45 *Let Γ be the set of κ-dense measures. Let μ_1 be the uniform measure over $\mathcal{P}$ of density κ ($\mu_1(x) = \kappa$ for all $x \in \mathcal{P}$), and let $|\mathcal{P}| = n$. Then for all $\mu \in \Gamma$ we have the following regret bound.*

$$\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_t, Q_t) \leq \frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \lambda + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n T}$$

Proof From Lemma 42, we have that

$$\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_{t+1}, Q_t) \leq \frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n T}$$

Adding $\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_t, Q_t)$ to both sides of the inequality and rearranging gives

$$\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_t, Q_t) \leq \frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \frac{1}{T}\sum_{t=1}^{T} \left(M(\hat{\mu}_t, Q_t) - M(\hat{\mu}_{t+1}, Q_t)\right) + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n T}.$$

We may then apply Lemma 44 to conclude

$$\frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}_t, Q_t) \leq \frac{1}{T}\sum_{t=1}^{T} M(\hat{\mu}, Q_t) + \lambda + \frac{\mathrm{KL}(\mu \parallel \mu_1)}{\lambda \kappa n T}.$$

■

Appendix K. Private Boosting with Stopping Rules

We now show that the `SmoothBoost` algorithm, due to Servedio (2003), can be adapted to fit in our private boosting framework. This discussion yields another differentially private halfspace learner and demonstrates that our framework is flexible enough to handle another "canonical" smooth boosting algorithm.

K.1. Boosting Schemas with Stopping Conditions

In Section 3.1, we decomposed boosting into its two most basic algorithmic parts: the weak learner `WkL` and the re-weighting strategy `NxM`. This assumed that T, the number of calls to the weak learner, is fixed in advance. But many boosting algorithms (including Servedio's SmoothBoost) employ a *stopping rule*: a test that determines adaptively whether or not to finish boosting and return the current aggregated hypothesis. Here, we formalize the common case where the stopping rule is a threshold function applied to the intermediate measure. In the boosting scheme of Section 3, the number of rounds is supplied as input. Below, the threshold against which the stopping rule is tested is supplied as input.

Just as in Section 3.1, we describe Boosting using a "helper function" for iteration. Again, this will make it easy to apply composition for differential privacy and enforce privacy for any boosting algorithm where `NxM` and `?END` satisfy certain minimal conditions, elaborated later. We denote by $\mathcal{H}$ the space of hypotheses used by the weak learner, S an iid sample from the target distribution $\mathcal{D}$, and $\mathcal{M}$ the set of bounded measures over S. Type signatures and the algorithmic schema follow.

$$\begin{aligned}
\text{Weak Learner, } \texttt{WkL} &: X^n \times \mathcal{M} \to \mathcal{H} \\
\text{Next Measure, } \texttt{NxM} &: X^n \times \mathcal{H}^* \to \mathcal{M} \\
\text{Stopping Criterion, } \texttt{?END} &: \mathcal{M} \to \mathbb{R} \\
\text{Iteration, } \texttt{Iter} &: X^n \times \mathcal{H}^* \to \mathcal{H}^*
\end{aligned}$$

Algorithm 10 `Boost`. Input: $S \in X^n$, $S \sim \mathcal{D}$, $\mathcal{T} \in \mathbb{R}$

```
$H \leftarrow \{\}$
repeat
    $H \leftarrow \texttt{Iter}(S, H, \mathcal{T})$
until $\perp \in H$
$T \leftarrow |H|$
$\hat{f}(x) \leftarrow \frac{1}{T}\sum_{i=1}^{T} h(x)$
return $\operatorname{sgn}(\hat{f}(x))$
```

Algorithm 11 `Iter`. Input: $S \in X^n$, $H \in \mathcal{H}^*$, $\mathcal{T} \in \mathbb{R}$

```
$\mu \leftarrow \texttt{NxM}(S, H)$
$\nu \sim \text{Lap}(1/\varepsilon)$
if $(\texttt{?END}(\mu) - \mathcal{T}) + \nu > 2 + \log(1/\delta)/\varepsilon$ then
    $h \leftarrow \texttt{WkL}(S, \mu)$
    return $H \circ h$
else
    return $\perp$ and terminate
end if
```

K.2. Approximate Differentially Private Learning

Our framework for handling boosting with stopping rules takes any weak learner satisfying *approximate* differential privacy and produces a strong learner satisfying this privacy notion as well.

Definition 46 (Differential Privacy) *A randomized algorithm $\mathcal{M} : X^n \to \mathcal{R}$ is (ε, δ)-differentially private if for all measurable $T \subseteq \mathcal{R}$ and all neighboring datasets $S \sim S' \in X^n$, we have*

$$\Pr[\mathcal{M}(S) \in T] \leq e^{\varepsilon} \Pr[\mathcal{M}(S') \in T] + \delta.$$

Definition 47 ((ε, δ)-Differentially Private Weak Learning) *A weak learning algorithm* `WkL` $:$ *$S \times \mathcal{D}(S) \to \mathcal{H}$ satisfies $(\varepsilon, \delta, \zeta)$-DP if for all neighboring samples $S \sim S' \in (\mathcal{X}^n \times \{\pm 1\})$, for all measurable $T \subseteq \mathcal{H}$, and any pair of distributions $\hat{\mu}, \hat{\mu}'$ on X such that $\Delta(\hat{\mu}, \hat{\mu}') < \zeta$ we have:*

$$\Pr[\mathtt{WkL}(S,\hat{\mu}) \in T] \leq e^{\varepsilon}\Pr[\mathtt{WkL}(S',\hat{\mu}') \in T] + \delta.$$

Again, we use composition theorems to handle the iterative structure of boosting. See Dwork et al. (2010); Bun and Steinke (2016) for the result below.

Theorem 48 (Composition for Approximate Differential Privacy) *Let $\varepsilon, \delta, \delta' > 0$. Then the k-fold adaptive composition of (ε,δ)-differentially private algorithms is $(\varepsilon', k\delta + \delta')$-differentially private for*

$$\varepsilon' = \frac{k\varepsilon^2}{2} + \varepsilon\sqrt{2k\log(1/\delta')}.$$

K.3. Ensuring Privacy with Stopping Rules

Under what circumstances will boosting algorithms using this new scheme satisfy privacy bounds? Just as in Section 3.2, the weak learner $\mathtt{WkL}$ must be $(\varepsilon,\delta,\zeta)$-DP (Definition 47). However, we relax the condition on the measure rule $\mathtt{NxM}$ to only be ζ-slick (Definition 19) *on measures which satisfy the guarantee checked by the stopping rule*.

Lemma 49 (Abstract Single-Round Privacy) *Suppose $\mathtt{WkL}$ is $(\varepsilon,\delta,\zeta)$-DP, $\mathtt{NxM}$ is ζ-slick under the promise that it is given a measure passing the end test $\mathtt{WkL}$, and that $\mathtt{END?}$ is a sensitivity-1 test. Then $\mathtt{Iter}$ run using these procedures is $(2\varepsilon, e^{\varepsilon}\delta)$-zCDP.*

Lemma 49 follows immediately from the following generalization of the Propose-Test-Release framework of Dwork and Lei (2009), taking the function f below to be $(?\mathtt{END}(\mu) - \mathcal{T})$.

Let $f : X^n \to \mathbb{R}$ be a sensitivity-1 function. Suppose A is a randomized algorithm with the following guarantee: If $S \sim S'$ and $f(S), f(S') \geq 0$, then for every measurable set T, we have $\Pr[A(S) \in T] \leq e^{\varepsilon}\Pr[A(S') \in T] + \delta$.

Lemma 50 *Let f and A be as above. Consider the following algorithm B: Let $\nu \sim \mathrm{Lap}(1/\varepsilon)$. If $f(S) + \nu \geq 2 + \log(1/\delta)/\varepsilon$, output $A(S)$. Otherwise, output $\perp$. Then B is $(2\varepsilon, e^{\varepsilon}\delta)$-differentially private.*

Proof Let $t = 2 + \log(1/\delta)/\varepsilon$. Fix neighboring datasets S, S'. First suppose $f(S) < 1$. Then by concentration properties of the Laplace distribution and the fact that f has sensitivity 1, we have $\Pr[f(S) + \nu \geq t] \leq \delta$ and $\Pr[f(S') + \nu \geq t] \leq \delta$. Therefore, $\Pr[B(S) = \perp] \geq 1 - \delta$ and $\Pr[B(S') = \perp] \geq 1 - \delta$ so $B(S), B(S')$ are (ε,δ)-indistinguishable.

Now suppose instead that $f(S) \geq 1$. Then since f has sensitivity 1, we also have $f(S') \geq 0$. Let T be a measurable subset of the range of A. Then we have

$$\begin{aligned}\Pr[B(S) \in T] &= \Pr[A(S) \in T] \cdot \Pr[f(S) + \nu \geq t]\\ &\leq (e^{\varepsilon}\Pr[A(S') \in T] + \delta) \cdot e^{\varepsilon}\Pr[f(S') + \nu \geq t]\\ &= e^{2\varepsilon}\Pr[B(S') \in T] + e^{\varepsilon}\delta.\end{aligned}$$

Similarly,

$$\begin{aligned}\Pr[B(S) \in T \cup \{\perp\}] &= \Pr[A(S) \in T] \cdot \Pr[f(S) + \nu \geq t] + \Pr[f(S) + \nu < t]\\ &\leq (e^{\varepsilon}\Pr[A(S') \in T] + \delta) \cdot e^{\varepsilon}\Pr[f(S') + \nu \geq t] + e^{\varepsilon}\Pr[f(S') + \nu < t]\\ &= e^{2\varepsilon}\Pr[B(S') \in T] + e^{\varepsilon}\delta.\end{aligned}$$

■

K.4. Servedio's SmoothBoost Algorithm

We do not state the whole SmoothBoost algorithm, but describe the sensitivity-1 end test and measure production function below. The following theorem summarizes its guarantees:

Theorem 51 (Servedio (2003)) *Suppose SmoothBoost is run with smoothness parameter $0 < \kappa < 1$ and weak learners which guarantee advantage $0 \leq \gamma < 1/2$. Then*

1. *The intermediate measures produced by SmoothBoost have density less than κ within $T < \frac{2}{\kappa\gamma^2\sqrt{1-\gamma}}$ rounds.*
2. *The final hypothesis output by SmoothBoost has margin at least $\theta = \gamma/(2+\gamma)$ on all but a κ fraction of inputs.*

Algorithm 12 `?END`(κn). Input: μ

```
if $\sum_{i=1}^{n} \mu(i) \leq \kappa n$ then
    return $\perp$
else
    return $\top$
end if
```

Algorithm 13 `SB-NxM`(κ): SmoothBoost Next Measure

Parameters: $\kappa \in (0,1)$, $0 \leq \theta \leq \gamma < 1/2$
Input: S, the sample; $H = \{h_1, \dots, h_t\}$, the sequence of hypotheses
Output: A measure over $[n]$, $n = |S|$

```
if $t = 0$ then
    $\mu_1(i) \leftarrow 1 \ \ \forall i \in [n]$
else
    $s_t(i) \leftarrow -t\theta + \sum_{\ell=1}^{t} y_i h_\ell(x_i) \ \ \forall i \in [n]$
    $\mu_{t+1}(i) \leftarrow \begin{cases} 1 & \text{if } s_t(i) < 0 \\ (1-\gamma)^{s_t(i)/2} & \text{if } s_t(i) \geq 0 \end{cases} \quad \forall i \in [n]$
end if
return $\mu_{t+1}$
```

K.5. Slickness of SmoothBoost

Lemma 52 (SmoothBoost Slickness) *Let μ, μ' be the measures produced when* `SB-NxM` *is run with the sequence of hypotheses $H = \{h_1, \dots, h_t\}$, and on neighboring datasets $S \sim S'$. Suppose that both measures μ, μ' are κ-dense. Then $\Delta(\hat{\mu}, \hat{\mu}') \leq 2/\kappa n$.*

Proof Let i be the index at which S and S' differ, and note that $\mu(j) = \mu'(j)$ for all $j \neq i$.

Let μ_0 denote the measure with $\mu_0(j) = \mu(j) = \mu'(j)$ for all $j \neq i$, and $\mu_0(i) = 1$. We will show that $\Delta(\hat{\mu_0}, \hat{\mu}) \leq 1/\kappa n$, which is enough to prove the claim by the triangle inequality. Note that $\mu_0(j) \geq \mu(j)$ for every j, and hence $|\mu_0| \geq |\mu| \geq \kappa n$.

We calculate

$$\begin{aligned}
\Delta(\hat{\mu_0}, \hat{\mu}) &= \frac{1}{2}\sum_{j=1}^{n}\left|\frac{\mu_0(j)}{|\mu_0|} - \frac{\mu(j)}{|\mu|}\right| \\
&= \frac{1}{2}\left(\left|\frac{1}{|\mu_0|} - \frac{\mu(i)}{|\mu|}\right| + \sum_{j\neq i}\left|\frac{\mu_0(j)}{|\mu_0|} - \frac{\mu(j)}{|\mu|}\right|\right) \\
&= \frac{1}{2}\left(\frac{2}{\kappa n} + \sum_{j\neq i}\mu(j)\left(\frac{1}{|\mu|} - \frac{1}{|\mu_0|}\right)\right) \\
&= \frac{1}{2}\left(\frac{2}{\kappa n} + \left|\frac{1}{|\mu|} - \frac{1}{|\mu|+1}\right|\right) \\
&= \frac{1}{2}\left(\frac{2}{\kappa n} + \frac{2}{\kappa n(\kappa n + 1)}\right) \\
&\leq \frac{2}{\kappa n}.
\end{aligned}$$

■

K.6. Putting Everything Together

Theorem 53 *There exists a constant c for which the following holds. Let $n \geq c\log(T/\beta\delta)/\varepsilon$. Suppose that* `WkL` *is an $(\varepsilon, \delta, 4/\kappa n)$-DP weak learner with advantage guarantee γ. Then with probability at least $1-\beta$,* `Boost` *run with* `WkL`*,* `SB-NxM`*(κ), and* `?END`*$(\kappa n/2)$ enjoys the following properties.*

1. `Boost` *runs for $T < \frac{4}{\kappa\gamma^2\sqrt{1-\gamma}}$ rounds.*
2. *For every $\delta' > 0$,* `Boost` *is $(\varepsilon', Te^{\varepsilon}\delta + \delta')$-DP for $\varepsilon' = 2T\varepsilon^2 + 2\varepsilon\sqrt{T\log(1/\delta')}$.*
3. *The final hypothesis H output by* `Boost` *has margin at least $\theta = \gamma/(2+\gamma)$ on all but a κ fraction of inputs.*

Proof Property 1 follows from the first claim of Theorem 51. After $T < \frac{2}{\kappa\gamma^2\sqrt{1-\gamma}}$ rounds, the measures produced by `SB-NxM` will have density less than κ. For sufficiently large n as guaranteed in the theorem statement, with probability at least $1-\beta$, the noisy end test will halt after the density decreases below κ, but before the density reaches $\kappa/2$.

Combining Lemma 49 with the slickness bound Lemma 52 implies that each call to `Iter` is $(2\varepsilon, e^{\varepsilon}\delta)$-differentially private. Property 3 now follows from the advanced composition Theorem 48.

Property 4 follows immediately from the second claim of Theorem 51, as our advantage assumption on `WkL` means our learner inherits the margin guarantee of SmoothBoost. ■

Appendix L. Direct Comparison to (Dwork et al., 2010)

We now compare our algorithm to that of Dwork et al. (2010). They start from an $(\varepsilon_b, \delta_b)$-DP weak learner with advantage γ and boost to training error at most $\hat{\alpha}$ under (ε, δ)-Differential Privacy. Their result is a combined privacy and sample bound, which we re-state as Theorem 56 for comparison's sake. By numerous techniques (including a simple appeal to differential privacy itself, see Appendix G) one could extend their training error bound to a generalization bound.

In this section, we hold utility and privacy objectives fixed under an identical weak learner assumption to that of Dwork et al. (2010). We show that `LazyBregBoost` saves a factor of γ^{-2} in sample complexity while recovering (essentially) the same training error and privacy guarantees (Theorem 57). Differences between our respective settings are summarized below.

Resampling vs. Weighted Boosting: There are two kinds of boosting implementations: *resamplers* use the intermediate distributions to draw a sample for the weak learner, while *weight boosters* supply sample weights explicitly to a weak learner. `Boost4People` boosts by resampling, while `LazyBregBoost` assumes the weak learner supports explicit sample weighting. Here, we give a resampling version of `LazyBregBoost` for comparison.

Slickness: The weak learner of Dwork et al. (2010) does not have a "slickness" property (as in our Definition 18) to enforce similar behavior on similar input distributions — which is necessary for privacy under boosting. They *impose* slickness on an arbitrary private weak learner by subsampling the input sample; this is one reason that `Boost4People` is a resampler booster. The resampling version of `LazyBregBoost` presented here also "automatically" imposes slickness.

Hypothesis Type: Our weak learner is real-valued, whereas the weak learner assumed by Dwork et al. (2010) is Boolean-valued. So, our setting is a slight generalization.

Aggregation Function: `LazyBregBoost` aggregates weak learners by averaging, whereas the `Boost4People` algorithm aggregates by majority vote. Thus, if a hypothesis class is closed under averaging, `LazyBregBoost` can learn *properly* (as for halfspaces, Theorem 26). Here, we aggregate by majority to more closely match Dwork et al. (2010).

L.1. Preliminaries

We begin by re-stating the weak learner assumption of Dwork et al. (2010).

Definition 54 ((m, γ, ξ)-Base Learner) *Let X be a universe, $\mathcal{C} = \{c : X \to \{\pm 1\}\}$ a concept class, and L a learning algorithm that receives m labeled items, each in $X \times \{\pm 1\}$ and outputs a hypothesis $h : X \to \{\pm 1\}$.*

We say that L is a (m, γ, ξ)-Base Learner for $\mathcal{C}$ if for every $c \in \mathcal{C}$, when the k items are drawn from a distribution D on X and labeled by c, with all but probability ξ over the choice of items and L's coins, the hypothesis h labels at least a $(1/2 + \gamma)$-fraction of the mass of D correctly. That is:

$$\Pr_{x \sim D}[h(x) = c(x)] \geq 1/2 + \gamma$$

We will alter our boosting schema to use resampling instead of explicit weighting. Accordingly, we require the following lemma of Dwork et al. (2010), which is used to enforce "slickness" for boost-by-resampling algorithms.

Lemma 55 (Lemma 6.5 of Dwork et al. (2010), Sampling to Amplify Privacy) *Let A and B be distributions on universe U with statistical distance at most ζ, and $\mathcal{M} : U^k \to V$ an (ε, δ)-DP private mechanism, where $k > 0$ and $\varepsilon \leq 1/2$. Then for every measurable $S \subseteq V$,*

$$\Pr[\mathcal{M}(A^k) \in S] \leq e^{4\varepsilon \cdot \zeta \cdot k} \Pr[\mathcal{M}(B^k) \in S] + \delta \cdot (1 + e^{2 \cdot \zeta \cdot k}).$$

L.2. Private Boosting by Resampling

Finally, we define a resampling version of LazyBregBoost, modifying the schema of Section 3.

Algorithm 14 `Boost-S`. In: $S \in X^n,\ T \in \mathbb{N}$

```
$H \leftarrow \{\}$
for $t = 1$ to $T$ do
    $H \leftarrow$ IterS$(S, H)$
end for
$\hat{f}(x) \leftarrow \mathrm{MAJ}_{i=1}^{T} h_i(x)$
return $\hat{f}(x)$
```

Algorithm 15 `IterS`. In: $S \in X^n,\ H \in \mathcal{H}^*$

```
$\mu \leftarrow$ LBNxM$(S, H)$
$S' \leftarrow$ Sample $k$ elements of $S$ using $\mu$
$h \leftarrow$ WkL$(S')$
return $H \cup \{h\}$
// Add $h$ to list of hypotheses
```

The sample bounds below should be compared: both start from an $(\varepsilon_b, \delta_b)$-DP weak learner with advantage γ. We hold training error $\hat{\alpha}$ and the parameter ε of approximate differential privacy fixed while allowing probability of learning failure and the δ parameter of approximate differential privacy to vary. Under these conditions, `LazyBregBoost` saves a factor γ^{-2} in sample complexity.

Theorem 56 (Theorem 6.1 of Dwork et al. (2010) using advanced composition) *Let L be a base learner for concept class $\mathcal{C}$. Then for any $\hat{\alpha}, \varepsilon, \delta', \upsilon \in (0, 1)$, when L is plugged into `Boost4People`, for appropriate settings of the parameters (and appropriate constants hidden in the $O(\cdot)$ notation) `Boost4People` guarantees:*

1. *If L is $(\varepsilon_b, \delta_b)$-DP then one round of `Boost4People` is $(\varepsilon^*, \delta_b(1 + \exp(\varepsilon^*))$-DP, where $\varepsilon^* = \varepsilon / \sqrt{8T \ln(1/\delta')}$.*

2. *By advanced composition (Theorem 48), it follows that T rounds of `Boost4People` satisfies $(\varepsilon,\ T \cdot \delta_b(1 + \exp(\varepsilon^*)) + \delta')$-DP.*

3. *If L is a (m, γ, β) base learner, then `Boost4People` runs for $T = O\left(\frac{\log(1/\hat{\alpha})}{\gamma^2}\right)$ rounds. For datasets of size at least*

$$n \in \Omega\left(\frac{\sqrt{T \ln(1/\delta')}\varepsilon_b \cdot m \cdot \upsilon \log(1/\hat{\alpha})}{\varepsilon \gamma^2 \hat{\alpha}}\right) = \Omega\left(\frac{\sqrt{\log(1/\hat{\alpha}) \ln(1/\delta')}\varepsilon_b \cdot m \cdot \upsilon \log(1/\hat{\alpha})}{\varepsilon \gamma^3 \hat{\alpha}}\right)$$

it guarantees that with all but $(T \cdot \beta + \exp(-\upsilon))$ probability, the output hypothesis has error at most $\hat{\alpha}$.

Theorem 57 *Let L be a base learner for concept class $\mathcal{C}$. Then for any $\hat{\alpha}, \delta', \varepsilon \in (0, 1)$, when L is plugged into `Iter-S`, for appropriate settings of the parameters (and appropriate constants hidden in the $O(\cdot)$ notation) `Boost-S` guarantees:*

1. *If L is $(\varepsilon_b, \delta_b)$-DP then one round of `Boost-S` satisfies $(\varepsilon^*, \delta_b(1 + \exp(\varepsilon^*)))$-DP, where $\varepsilon^* = \varepsilon / \sqrt{8T \ln(1/\delta')}$.*

2. *By advanced composition (Theorem 48), it follows that T rounds of* `Boost-S` *satisfies* $(\varepsilon,\ T \cdot \delta_b \cdot (1 + \exp(\varepsilon^*)) + \delta')$*-DP.*

3. *If L is a (m, γ, β) base learner, then* `Boost-S` *runs for $T = O\left(\frac{\log(1/\hat{\alpha})}{\gamma^2}\right)$ rounds. For datasets of size at least*

$$n \in \Omega\left(\frac{\sqrt{T \ln(1/\delta')} \cdot \varepsilon_b m}{\varepsilon \hat{\alpha}}\right) = \Omega\left(\frac{\sqrt{\log(1/\hat{\alpha}) \ln(1/\delta')}\varepsilon_b m}{\varepsilon \gamma \hat{\alpha}}\right)$$

it guarantees that with all but $(T \cdot \beta)$ probability, the output hypothesis has error at most $\hat{\alpha}$.

For utility, we appeal to Theorem 20, the round bound for boosting with `LB-NxM`. From the base learner L, a weak learner L' as in our Definition 11 can be produced by first sampling k examples from the target distribution D and then invoking L. Thus, setting $\kappa \in \Theta(\hat{\alpha})$, we can achieve training error at most $\hat{\alpha}$ when boosting concludes. To ensure there are enough examples for each run of the base learner, we require $|S| = n > m$.

For the privacy analysis, we re-express the strategy above by "lifting" the sampler out of the weak learner and into the inner loop of boosting. This scheme will behave identically to boosting with L' — as can be seen by comparing `Boost-S` and `Iter-S` to their counterparts in Section 3 — but is better partitioned to apply privacy composition lemmas.

By Lemma 21, `LB-NxM` has statistical distance between D_t, D'_t induced by neighboring samples S and S' at most $1/(\kappa n)$. Using this bound on statistical distance and arguing as in Lemma 6.3, item 3 of Dwork et al. (2010), we get that `Iter-S` invoked with a Base Learner parameterized as above is

$$\left(\frac{4\varepsilon_b m}{\kappa n},\ \delta_b\left(1 + \exp\left(\frac{2\varepsilon_b m}{\kappa n}\right)\right)\right)$$

differentially private. By advanced composition over T rounds of boosting, this is

$$\left(\sqrt{T \ln(1/\delta')} \cdot \frac{8\varepsilon_b m}{\kappa n},\ T \cdot \delta_b\left(1 + \exp\left(\frac{2\varepsilon_b m}{\kappa n}\right)\right) + \delta'\right)$$

differentially private overall, for any $\delta' \in (0,1)$. Given a privacy target ε, we want:

$$\sqrt{T \ln(1/\delta')} \cdot \frac{8\varepsilon_b m}{\kappa n} \leq \varepsilon$$

Setting $\kappa \in \Theta(\hat{\alpha})$, we have:

$$\varepsilon \in \Omega\left(\sqrt{T \ln(1/\delta')} \cdot \frac{\varepsilon_b m}{\hat{\alpha} n}\right)$$

which lower-bounds the sample size:

$$n \in \Omega\left(\sqrt{T \ln(1/\delta')} \cdot \frac{\varepsilon_b m}{\hat{\alpha} \varepsilon}\right)$$

Expanding T and substituting gives the claimed sample and privacy bounds for `Boost-S`.