On PAC Learning Halfspaces in Non-interactive Local Privacy Model with Public Unlabeled Data

Jinyan Su JINYAN.SU@MBZUAI.AC.AE
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

Jinhui Xu JINHUI@BUFFALO.EDU
Department of Computer Science and Engineering
State University of New York at Buffalo

Di Wang DI.WANG@KAUST.EDU.SA
Division of CEMSE
Computational Bioscience Research Center
SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence
King Abdullah University of Science and Technology (KAUST)

Editors: Emtiyaz Khan and Mehmet Gonen

Abstract

In this paper, we study the problem of PAC learning halfspaces in the non-interactive local differential privacy model (NLDP). To breach the barrier of exponential sample complexity, previous results studied a relaxed setting where the server has access to some additional public but unlabeled data. We continue in this direction. Specifically, we consider the problem under the standard setting instead of the large margin setting studied before. Under different mild assumptions on the underlying data distribution, we propose two approaches that are based on the Massart noise model and self-supervised learning and show that it is possible to achieve sample complexities that are only linear in the dimension and polynomial in other terms for both private and public data, which significantly improve the previous results. Our methods could also be used for other private PAC learning problems. [1]

Keywords: Differential privacy; PAC learning; Learning halfspaces.

1. Introduction

A tremendous quantity of sensitive data is generated and gathered every day. Due to the sensitive information of these data, how to enable the benefit of analyzing the data without exposing the individual information has become an important issue. To address the issue, *Differential Privacy* (DP) Dwork et al. (2006) has become as the de facto tool for privacy-preserving data analysis. There are two well-studied models in DP- the *central* model and the *local* model. In the central model, the raw data is collected by a central server and then processed by a DP algorithm while in the local model Evfimievski et al. (2003), each individual applies a DP algorithm locally and sends only the output of the algorithm to the server. Local model is used more often when learning in a distributed system or when users do not trust the central data collector.

1. Part of the work was done when Jinyan Su was a research intern at KAUST.

In the local differential privacy (LDP) model, the communication between the server and individual users could be either in one round or in multiple rounds, and these two communication protocols of LDP are called non-interactive LDP (NLDP) or interactive LDP correspondingly. However, in practice, NLDP is preferred over interactive LDP because of the latency and waiting for responses takes a large amount of time, and thus it is necessary to limit the number of interactions. Moreover, current deployments of LDP algorithms are all non-interactive protocols, such as Google and Apple Cormode et al. (2018); Tang et al. (2017); Erlingsson et al. (2014); Near (2018).

Beginning from Kasiviswanathan et al. (2011), there is a long list of work studying the Valiant's probabilistically approximately correct (PAC) learning model Valiant (1984) under DP constraint and what concepts we can learn privately, such as Blum et al. (2013); Bun et al. (2020). While private PAC learning is well studied in the central DP model and interactive LDP model, its theoretical behaviors in the NLDP model are much more challenging and are still far from well-understood due to the restriction on the number of rounds of communication. Daniely and Feldman (2019) provided the first study of the problem and proved that only classes that have polynomially small margin complexity can be efficiently PAC learned by an NLDP algorithm. Recently, Dagan and Feldman (2020) studied the PAC learning halfspaces in NLDP model. While halfspaces is PAC learnable in the central DP model and interactive LDP model Lê Nguyên et al. (2020); Beimel et al. (2019); Kasiviswanathan et al. (2011), unfortunately, Dagan and Feldman (2020) showed that even for learning halfspaces under large-margin assumptions requires an exponential number of samples in the NLDP model, which indicates that in general, halfspaces is unlearnable in NLDP model. To breach the barrier of exponential sample complexity, Daniely and Feldman (2019) studied a relaxed NLDP model where the server is allowed to access some public but unlabeled data. Specifically, they considered the large margin setting and showed the following result (see Section 3 for the definitions of large margin setting and NLDP Learner).[2]

Theorem 1 *Daniely and Feldman (2019) Under the large margin setting, there is a computationally efficient $(\epsilon, \alpha, \beta, \gamma)$-NLDP Learner with sample complexity $n = \widetilde{O}(\frac{d^{10}\log(1/\beta)}{\epsilon^2 \cdot \gamma^{12}\alpha^6})$ for private data and $m = \widetilde{O}(\frac{d^{10}\log(1/\beta)}{\epsilon^2 \cdot \gamma^{12}\alpha^6})$ for public unlabeled data, where d is the dimension of the space, γ is the margin, α is the target error and β is the failure probability.*

However, there are two issues with the result. First, Theorem 1 only holds for the large margin setting, which is stronger than the standard (non-large margin) setting. Secondly, compared with the non-private case where the sample complexity is only linear on $\frac{1}{\gamma}$ and is independent on d Shalev-Shwartz and Ben-David (2014), the dependencies on d, $\frac{1}{\gamma}$ in Theorem 1 are unsatisfactory. In this paper, we revisit the problem and partially address these issues. Specifically, we consider PAC learning halfspaces in the NLDP model under the standard setting and show that it is possible to achieve a sample complexity that is only **linear** in d (and polynomial in other terms) for both private and public data, if the underlying data distribution satisfies some mild assumptions. Our contributions can be summarized as follows.

1. We first study the case where the data distribution satisfies the anti-anti-concentration and anti-concentration properties. We propose an (ϵ, δ)-NLDP algorithm which is motivated by the Massart noise learning model and show that its sample complexity to achieve the error α is $\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon}, \frac{1}{\alpha}))$ and $O(\frac{d}{\alpha^4})$ for private and public data respectively.

2. Since in Daniely and Feldman (2019) did not provide the explicit form of the sample complexities, in Theorem 1 we rewrite their result, see Appendix for its proof.

Methods	Sample Complexity	Measure	Loss Function	With public data?	Data
Smith et al. (2017)	$O(d\epsilon^{-2}\alpha^{-2})$	Excess Risk	Linear Regression	No	ℓ_2-norm Bounded
Smith et al. (2017)	$\widetilde{O}(2^d\alpha^{-(d+1)}\epsilon^{-2})$	Excess Risk	Lipschitz and Convex	No	ℓ_2-norm Bounded
Wang et al. (2018)	$\widetilde{O}(4^{d(d+1)}D_d^2\epsilon^{-2}\alpha^{-4})$	Excess Risk	(∞, T)-smooth	No	ℓ_2-norm Bounded
Wang et al. (2019, 2020)	$d\cdot\left(\frac{C}{\alpha^3}\right)^{O(1/\alpha^3)}/\epsilon^{O(\frac{1}{\alpha^3})}$	Excess Risk	Lipschitz Convex GLM	No	ℓ_2-norm Bounded
Zheng et al. (2017)	$d(\frac{8}{\alpha})^{O(\log\log(\frac{1}{\alpha}))}(\frac{4}{\epsilon})^{O(\log(\frac{1}{\alpha}))}$	Excess Risk	Convex ∞-Smooth GLM	No	ℓ_2-norm Bounded
Wang et al. (2021)	$O(d^3\alpha^{-2}\epsilon^{-2})$	ℓ_2-norm Error	Smooth GLM	Yes, $O(\frac{d}{\alpha^2})$	Gaussian
Wang et al. (2021)	$O(d^2\alpha^{-2}\epsilon^{-2})$ for $\alpha\geq\Omega(\frac{1}{\sqrt{d}})$	ℓ_∞-norm Error	Smooth GLM	Yes, $O(\frac{d}{\alpha^2})$	ℓ_1-norm Bounded and Sub-Gaussian
Daniely and Feldman (2019)	$\widetilde{O}(\frac{d^{10}}{\epsilon^2\cdot\gamma^{12}\alpha^6})$	Excess Risk	0-1 loss/large margin halfspace	Yes, $\widetilde{O}(\frac{d^{10}}{\epsilon^2\cdot\gamma^{12}\alpha^6})$	ℓ_2-norm Bounded
This Paper	$\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon},\frac{1}{\alpha}))$	Excess Risk	0-1 loss/ halfspace	Yes, $O(\frac{d}{\alpha^4})$	Structured distribution
This Paper	$\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon},\frac{1}{\alpha}))$	Excess Risk	0-1 loss/halfspace	Yes, $\widetilde{O}(\frac{d}{\alpha^2})$	Structured distribution

Table 1: Comparisons on the sample complexities for private and public unlabeled data to achieve error α under different measurements and assumptions, where C is a constant and D_d is a function of d, γ is the margin of the data. For bounded norm case we assume that $\|x_i\|\leq R=O(1)$ for every $i\in[n]$. We also assume the loss functions are Lipschitz.

2. To further reduce the sample complexity of public data, we then study the case where the underlying distribution follow a mixture distribution and show that it is possible to achieve sample complexity of $\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon},\frac{1}{\alpha}))$ and $\widetilde{O}(\frac{d}{\alpha^2})$ for private and public data respectively. Instead of the Massart noise model, our algorithm is motivated by self-supervised learning.

Due to the space limit, all proofs and omitted algorithms are included in Appendix.

2. Related Work

As mentioned, although there are numerous results on private PAC learning halfspaces, the problem in the NLDP model with public unlabeled data has only been studied by Daniely and Feldman (2019). However, it differs from our results in quite a few ways. Firstly, their algorithm considered a large margin setting and can not be applied to the general setting which is studied in this paper. Secondly, although both their work and ours used public unlabeled data, the usage of these data is quite different. Daniely and Feldman (2019) used the public unlabeled data to compute the gradient of the decomposed gradient while we use algorithms to label the public data and conduct the learning process on the public data. Finally, Daniely and Feldman (2019) studied Data-Independent PAC learning while we focus on Data-Dependent PAC learning. Thus, our sample complexities are lower than theirs.

Besides PAC learning, recently there are several works studied the problem of Stochastic Convex Optimization in NLDP model (without public data), such as Smith et al. (2017); Wang et al. (2018,

2019, 2020); Zheng et al. (2017), see Table 1 for a summary. However, as we can see from Table 1, all of these methods need to assume the loss function is smooth enough and the sample complexities of these methods are exponential in d or the error α. Thus, these methods cannot be used for our problem as our loss is $0-1$ loss and we aim to make the sample complexity to be polynomial. To remedy the exponential sample complexity, Wang et al. (2021) studied the Generalized Linear Model in NLDP model with public unlabeled data. However, they need to assume that the loss function is smooth and the polynomial sample complexity only holds when the error α is not small enough. While our results hold for any $\alpha \in (0,1)$. Moreover, the usage of the public data is quite different.

3. Preliminaries

In this section, we will introduce some notations in PAC learning halfspaces and differential privacy.
Notations: Throughout the whole paper, we denote $\mathcal{P}$ as a probability distribution over $\mathcal{X} \times \{\pm 1\}$ with marginal distribution $\mathcal{P}_x$ over $\mathbb{R}^d$, where $\mathcal{X} \subseteq \mathbb{R}^d$. We also denote $\mathcal{B}_2^d(R)$ as the ℓ_2-norm ball in $\mathbb{R}^d$ with center 0 and radius R and $\mathcal{B}_2^d = \mathcal{B}_2^d(1)$ as the unit ℓ_2-norm ball. Given a convex constraint set $\mathcal{C} \subseteq \mathbb{R}^d$ and a loss function $\ell : \mathcal{C} \times (\mathcal{X} \times \{\pm 1\})$, we denote the population risk function as $L_{\mathcal{P}}(w) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(w;x,y)]$. Moreover, given an n-size dataset $D = \{(x_1,y_1),\cdots,(x_n,y_n)\} \sim \mathcal{P}^n$ we denote the empirical risk function of the loss over D, $\hat{L}(\cdot, D)$, as $\hat{L}(w,D) = \frac{1}{n}\sum_{i=1}^{n} \ell(w;x_i,y_i)$.

3.1. PAC Learning Halfspaces

In this paper we mainly focus on PAC (probably approximately correct) learning model Valiant (1984) for halfspaces in the realizable setting. That is, for any sample $(x,y) \sim \mathcal{P}$ we assume that $y = \text{sign}(\langle w^*, x\rangle + \theta^*)$ (almost surely) for some unknown vector $w^* \in \mathbb{R}^d$ and $\theta^* \in \mathbb{R}$. Without loss of generality we assume that $\theta^* = 0$ so $y = \text{sign}(\langle w^*, x\rangle)$. A linear threshold function is defined as $f_w(x) = \text{sign}(\langle w, x\rangle)$ where $x, w \in \mathbb{R}^d$ and we call the vector w a hypothesis, and the classification error of hypothesis w is

$$\begin{aligned} err_{\mathcal{P}}(f_w) &= \Pr_{(x,y)\sim\mathcal{P}}[f_w(x) \neq y] = \Pr_{(x,y)\sim\mathcal{P}}[\text{sign}(\langle w, x\rangle) \neq y] \\ &= \Pr_{(x,y)\sim\mathcal{P}}[y \cdot \langle w, x\rangle < 0]. \end{aligned}$$

Given $\alpha, \beta \in (0,1)$, the goal of PAC learning halfspaces is to find a hypothesis $w \in \mathbb{R}^d$ such that $err_{\mathcal{P}}(f_w) \leq \alpha$ with probability at least $1-\beta$ with low sample complexity. In the following we will introduce both the standard setting and the large margin setting.

Standard setting: Here we assume without loss of generality that $\mathcal{X} \subset \mathcal{B}_2^d(R)$ with some constant $R = O(1)$ and $w^* \in \mathbb{R}_2^d$. Formally, we aim to design an (α,β)-PAC learner.

Definition 2 ((α,β)-PAC learner) *Let $\mathcal{P}$ be a distribution over $\mathcal{B}_2^d(R) \times \{\pm 1\}$ such that there exists $w^* \in \mathcal{B}_2^d$ which satisfies $Pr_{(x,y)\sim\mathcal{P}}[y\langle w^*, x\rangle \geq 0] = 1$. We say an algorithm $\mathcal{A}$ an (α,β)-PAC learner with sample complexity n if using a dataset $D \sim \mathcal{P}^n$, the output classifier $\hat{w} = \mathcal{A}(D) \in \mathcal{B}_2^d$ satisfies $\Pr_{(x,y)\sim\mathcal{P}}[y \neq sign(\langle \hat{w}, x\rangle)] \leq \alpha$ with probability at least $1-\beta$.*

Large margin setting: Compared with the standard setting, in the large margin setting we additionally assume there is no example that falls too close to the boundary of the halfspace. Specifically, we assume that $\mathcal{X} \subset \mathcal{B}_2^d(R)$ with some constant $R = O(1)$ and $w^* \in \mathbb{R}_2^d$. Moreover, we assume that w^* maximizes the margin $\gamma = \min_{(x,y)\sim\mathcal{P}} \frac{|\langle w^*, x\rangle|}{||w^*||_2 \cdot ||x||_2} > 0$, which is known in advance. Under this setting we want to design an (α, β, γ)-PAC learner.

Definition 3 ((α, β, γ)-PAC learner) *Let $\mathcal{P}$ be a distribution over $\mathcal{B}_2^d(R) \times \{\pm 1\}$ such that there exists $w^* \in \mathcal{B}_2^d$ which satisfies $Pr_{(x,y)\sim\mathcal{P}}[y\langle w^*, x\rangle \geq \gamma] = 1$, then we call $\mathcal{P}$ a distribution with margin γ. We say an algorithm $\mathcal{A}$ an (α, β, γ)-PAC learner with margin γ and sample complexity n if using a dataset $D \sim \mathcal{P}^n$ with margin γ, the output classifier $\hat{w} = \mathcal{A}(D) \in \mathcal{B}_2^d$ satisfies the $\Pr_{(x,y)\sim\mathcal{P}}[y \neq sign(\langle \hat{w}, x\rangle)] \leq \alpha$ with probability at least $1 - \beta$.*

3.2. Differential Privacy

Definition 4 (Differential Privacy Dwork et al. (2006)) *Given a data universe $\mathcal{D}$, we say that two datasets $D, D' \subseteq \mathcal{D}^n$ are neighbors if they differ by only one entry, which is denoted as $D \sim D'$. A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private (DP) if for all neighboring datasets D, D' and all output event E of algorithm $\mathcal{A}$, we have $Pr(\mathcal{A}(D) \in E) \leq e^\epsilon Pr(\mathcal{A}(D') \in E) + \delta$. If $\delta = 0$, we say that algorithm $\mathcal{A}$ is ϵ-DP.*

Differential privacy in the local model. In LDP, we have a data universe $\mathcal{D}$, n players with each holding a private data record $x_i \in \mathcal{D}$, and a server coordinating the protocol. An LDP protocol executes a total of T rounds. In each round, the server sends a message, which is also called a query, to a subset of the players requesting them to run a particular algorithm. Based on the query, each player i in the subset selects an algorithm $\mathcal{A}_i$, runs it on her own data, and sends the output back to the server.

In NLDP model, we consider the distributed setting with star network. And each user has only one data sample. He/she needs to privatize his/her message before sending to the sever, and then the server aggregate these private information to perform analysis. Unlike the federated setting, since each user only has one sample, he/she cannot compute the target locally. And this is the main difficulty of learning in the NLDP model.

Definition 5 (Dwork et al. (2006)) *An algorithm $\mathcal{A}$ is (ϵ, δ)-locally differentially private (LDP) if for all pairs $x, x' \in \mathcal{D}$, and for all events E in the output space of $\mathcal{A}$, we have $Pr[\mathcal{A}(x) \in E] \leq e^\epsilon Pr[\mathcal{A}(x') \in E] + \delta$. A multi-player protocol is (ϵ, δ)-LDP if for all possible inputs and runs of the protocol, the transcript of player i's interaction with the server is (ϵ, δ)-LDP. If $T = 1$, we say that the protocol is ϵ non-interactive LDP (NLDP). When $\delta = 0$, we call it ϵ-NLDP.*

As we mentioned previously, PAC learning halfspaces in the NLDP model requires the sample complexity which is at least exponential in the dimension d even in the large margin setting Dagan and Feldman (2020). Thus, inspired by this, instead of the NLDP model, in this paper we will mainly focus on a relaxed NLDP model.

Our Model: Different from the above classical NLDP model where only one private dataset $D = \{(x_i, y_i)\}_{i=1}^n$ exists, the NLDP model in our setting allows the server to have an additional public unlabeled dataset $D' = \{q_j\}_{j=1}^m \subset \mathcal{X}^m$, where each q_j is sampled from $\mathcal{P}_x$, which is the marginal distribution of $\mathcal{P}$.

Thus, we aim to design some private (α, β) or (α, β, γ)-PAC learner in the NLDP model with public but unlabeled data. Moreover, we want the sample complexity of private data and public data to be as low as possible.

Definition 6 ($(\epsilon, \delta, \alpha, \beta)$-NLDP Learner) *Let $\mathcal{P}$ be a distribution over $\mathcal{B}_2^d(R) \times \{\pm 1\}$ such that there exists $w^* \in \mathcal{B}_2^d$ which satisfies $Pr_{(x,y)\sim\mathcal{P}}[y\langle w^*, x\rangle \geq 0] = 1$. We call an algorithm $\mathcal{A}$ an $(\epsilon, \delta, \alpha, \beta)$-NLDP PAC learner with sample complexity (n, m) if using a (private) dataset $D \sim \mathcal{P}^n$ and a public but unlabeled dataset $D' \sim \mathcal{P}_x^m$, the output classifier $\hat{w} = \mathcal{A}(D, D') \in \mathcal{B}_2^d$ satisfies the following with probability at least $1 - \beta$, $\Pr\limits_{(x,y)\sim\mathcal{P}}[y \neq sign(\langle \hat{w}, x\rangle)] \leq \alpha$. Moreover, the algorithm $\mathcal{A}$ is (ϵ, δ)-NLDP w.r.t the private dataset.*

Definition 7 ($(\epsilon, \delta, \alpha, \beta, \gamma)$-NLDP Learner) *Let $\mathcal{P}$ be a distribution over $\mathcal{B}_2^d(R) \times \{\pm 1\}$ such that there exists $w^* \in \mathcal{B}_2^d$ which satisfies $Pr_{(x,y)\sim\mathcal{P}}[y\langle w^*, x\rangle \geq \gamma] = 1$. We call an algorithm $\mathcal{A}$ an $(\epsilon, \delta, \alpha, \beta)$-NLDP PAC learner with sample complexity (n, m) if using a (private) dataset $D \sim \mathcal{P}^n$ and a public but unlabeled dataset $D' \sim \mathcal{P}_x^m$, the output classifier $\hat{w} = \mathcal{A}(D, D') \in \mathcal{B}_2^d$ satisfies the following with probability at least $1 - \beta$, $\Pr\limits_{(x,y)\sim\mathcal{P}}[y \neq sign(\langle \hat{w}, x\rangle)] \leq \alpha$. Moreover, the algorithm $\mathcal{A}$ is (ϵ, δ)-NLDP w.r.t the private dataset.*

Since any (ϵ, δ)-NLDP algorithm can be transformed to an ϵ-NLDP algorithm with almost the same accuracy Bun et al. (2019), here we only focus on (ϵ, δ)-NLDP for simplicity.

4. NLDP Algorithm via Massart noise model

Before showing our algorithm, we first introduce the Massart noise model in PAC learning:

Definition 8 (Massart noise example oracle Massart and Nédélec (2006)) *Let $\mathcal{S}$ be a concept class of Boolean functions over $\mathbb{R}^d$, $\mathcal{F}$ be a known family of structured distributions on $\mathbb{R}^d$, and let f be an unknown target function in $\mathcal{S}$. Assume $0 < \lambda < \frac{1}{2}$, a Massart noise example oracle $EX^{Mas}(f, \mathcal{F}, \lambda)$ is an oracle that each invoke returns a labeled example (x, y) such that:*

1. *$x \sim \mathcal{P}_x$, where $\mathcal{P}_x$ is a fixed distribution in $\mathcal{F}$.*
2. *With probability $1 - \lambda(x)$, the oracle returns the correct label $y = f(x)$ and with probability $\lambda(x)$, the oracle returns a misleading label $y = -f(x)$, i.e., $y = \begin{cases} f(x), \text{w.p. } 1 - \lambda(x) \\ -f(x), \text{w.p. } \lambda(x) \end{cases}$, where $\lambda(x)$ is unknown and satisfies $\lambda(x) \leq \lambda$.*

We can think of the Massart noise model as an adversary who flips each sample label independently with probability **at most** $\lambda < \frac{1}{2}$ and the target of PAC learner is to reconstruct the classifier to arbitrarily high accuracy.

Definition 9 (PAC Learning with Massart Noise) *Denote $\mathcal{P}$ the joint distribution on (x, y) generated by a Massart noise example oracle. The goal of PAC Learning with Massart Noise is to design an algorithm such that given i.i.d. samples from $\mathcal{P}$, the algorithm outputs a hypothesis h such that $Pr_{x\sim\mathcal{P}_x}[h(x) \neq f(x)] \leq \alpha$ with probability at least $1 - \beta$.*

Algorithm 1 NLDP based on Massart noise model

Input: Private data $D = \{(x_i, y_i)\}_{i=1}^n$ with each $x_i \in \mathbb{R}^d$ satisfying $||x_i||_2 \leq R$ and $y_i \in \{\pm 1\}$; Unlabeled public data $D' = \{q_i\}_{i=1}^m$; private parameters ϵ, δ; oracle access to Hinge Loss-LDP $\mathcal{H}_{priv}$ (Algorithm 5); error bound α; failure probability β.
Randomly divide n private data record into k groups: $\{S_1, \cdots, S_k\}$, where $|S_i| = \lfloor \frac{n}{k} \rfloor$, $k = O(\log \frac{1}{\beta})$.
for $t \in [k]$ **do**
    Denote $\widetilde{S}_t$ as the normalized version of S_t, *i.e.,* $\widetilde{S}_t = \{(\frac{x}{R}, y) | (x, y) \in S_t\}$.
    Set $w_t = \mathcal{H}_{priv}(\frac{1}{32R}, \epsilon, \delta, \widetilde{S}_t)$.
    Set $h_{w_t}(x) = \text{sign}(w_t^T x)$
end for
Get the Massart Noise example oracle by majority voting: $\hat{f}(x) = \arg\min_{y \in \{\pm 1\}} \sum_{t=1}^k \mathbb{I}(h_{w_t}(x) \neq y) = \arg\min_{y \in \{\pm 1\}} \sum_{t=1}^k \mathbb{I}(\text{sign}(w_t^T x) \neq y)$
for $i \in [m]$ **do**
    Label public dataset $\{q_i\}_{i=1}^m$ using Massart Noise example oracle $\hat{f}(x)$ to obtain the noisy dataset $\hat{D} = \{(q_i, \hat{f}_i)\}_{i=1}^m$, where $\hat{f}_i = \hat{f}(q_i)$.
end for
Run the subroutine LHMN$(\alpha, \beta, (U, r, R))$ with dataset $\hat{D}$ to get $\hat{w}$ =LHMN$(\alpha, \beta, (U, r, R))$.
Return $\hat{w}$ and $h_{\hat{w}} = \text{sign}(\hat{w}^T x)$.

Massart noise model lies in between the Random Classification Noise Angluin and Laird (1988) (where each label is independently flipped with probability **exactly** $\lambda \leq \frac{1}{2}$) and the agnostic model Kearns et al. (1994) (where an adversary can flip any small constant fraction of the sample labels) and has attracted much attention in recent years. Many algorithms for computing accurate hypothesis in the distribution-specific PAC learning has been promoted, such as Awasthi et al. (2015, 2016); Zhang et al. (2017). Recently, an efficient and simple algorithm has been proposed in Diakonikolas et al. (2020), which succeeds under more general distributional assumptions.

Definition 10 (Diakonikolas et al. (2020)) *Fix $U, r > 0$. An isotropic (i.e., zero mean and identity covariance) distribution $\mathcal{P}_x$ on $\mathbb{R}^d$ satisfies U-anti-concentration (2-dim) if for any projection $(\mathcal{P}_x)_V$ of $\mathcal{P}_x$ onto a 2-dimensional subspace V, the corresponding probability density function γ_V on $\mathbb{R}^2$ satisfies that for all $x \in V$, $\gamma_V(x) \leq U$. Moreover, we say (U, r)-anti-anti-concentration holds if for all $x \in V$ such that $||x||_2 \leq r$, $\gamma_V(x) \geq \frac{1}{U}$.*

Anti-anti-concentration and anti-concentration are mild distributional conditions about the probability density function on the projected 2-dimensional subspace. The former guarantees that at least a constant probability mass is assigned to the points near the origin of the projected 2-dimensional subspace while the latter states that the probability mass along the 2-dimensional projection is upper bounded.

In fact, several reasonable distribution families satisfy the previous two conditions. For example, the class of isotropic log-concave distribution satisfies (U, r)-anti-anti-concentration and U-anti-concentration with $U, r = \Theta(1)$ (See Fact A.4 in Diakonikolas et al. (2020)). Moreover, any isotropic s-concave distribution on $\mathbb{R}^d$ with $s \geq -\frac{1}{2d+3}$ satisfies (U, r)-anti-anti-concentration and U-anti-concentration with $U, r = \Theta(1)$ (See Appendix A.4 in Diakonikolas et al. (2020)).

Next we will present our Non-interactive LDP algorithm via the Massart noise model (Algorithm 1). Generally, the algorithm consists of two parts:

(1) First, we use private data to construct a Massart noise example oracle with rate $\lambda = \frac{3}{16}$. To get the oracle, in Algorithm 1 we first randomly divide the private data into $k = O(\log \frac{1}{\beta})$ disjoint groups. Then, on each group of data S_i, we consider the Empirical Risk Minimization problem with the hinge loss $\ell(w; x, y) = \max\{0, 1 - y\langle w, x\rangle\}$ with $\mathcal{C} = \mathcal{B}_2^d$. Specifically, when $n = \widetilde{O}\left(kd \log(\frac{1}{\beta})\text{Poly}(\log \frac{1}{\delta}, \frac{1}{\epsilon})\right)$, we can use an (ϵ, δ)-NLDP algorithm $\mathcal{H}_{priv}$ given by Wang et al. (2020) to get private estimator w_i such that

$$\mathbb{E}[\hat{L}(w_i, S_i)] - \min_{||w||_2 \leq 1} \hat{L}(w, S_i) \leq \frac{1}{32},$$

where $\hat{L}(w, S) = \frac{1}{|S|} \sum_{(x,y)\in S} \ell(w; x, y)$. After getting private estimators $\{w_i\}_{i=1}^k$, we then boost the classification accuracy using the majority voting mechanism. We can show that the new classifier $\hat{f}$ via voting is a Massart noise example oracle ($\lambda = \frac{3}{16}$) with probability at least $1 - \beta$.

(2) We then label $m = O(\frac{U^{12}}{r^{12}} \cdot \frac{d}{\alpha^4})$ public unlabeled data samples $D' = \{q_i\}_{i=1}^m$ with the learned Massart noise example oracle $\hat{f}$ and denote the labels as $\{\hat{f}_i\}_{i=1}^m$, where U, r are the parameters of anti-anti-concentration and anti-concentration in Definition 10. Then, we can invoke efficient and non-private algorithm LHMN designed for leaning halfspaces with Massart noise (Algorithm 2) on dataset $\hat{D} = \{(q_i, \hat{f}_i)\}_{i=1}^m$ to finally learn a classifier with any desired classification error α with probability at least $1 - \beta$. Formally, Algorithm 1 has the following theoretical guarantee.

Algorithm 2 Learning Halfspaces with Massart Noise : LHMN $(\alpha, \beta, (U, r, R))$

Input: The designed estimation error α; parameters about the distribution: U, r, R; failure probability β; loss function $g(w; (x, y)) = S_\sigma(-y\frac{\langle w,x\rangle}{||w||_2})$, where $S_\sigma(t) = (1 + e^{-\frac{t}{\sigma}})^{-1}$, dataset $\hat{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$ labeled by Massart noise example oracle with $\lambda = \frac{3}{16}$.

Set $C_1 = \Theta(\frac{U^{12}}{r^{12}})$, $C_2 = \Theta(\frac{r}{U^2})$, $T = \Theta(\frac{C_1 dR^8 \log(\frac{1}{\beta})}{\alpha^4})$, $\sigma = \frac{C_2\alpha}{\sqrt{2}R^2}$

Set step size $\eta = \frac{C_2^2 d\alpha^2}{8R^4 T^{1/2}}$.

Set $w^{(0)} = \boldsymbol{e_1}$ with $\boldsymbol{e_1}$ being the unit vector whose first component is 1, and other components are 0.

for $i = 1, \cdots, T$ **do**

  $v^{(i)} = w^{(i-1)} - \eta \nabla_w g(w^{(i-1)}; (x^{(i)}, y^{(i)}))$

  $w^{(i)} = \frac{v^{(i)}}{||v^{(i)}||_2}$

end for

Set the list of candidate vector $L = \{\pm w^{(i)}\}_{i\in[T]}$

Draw $N = O(\frac{\log(\frac{T}{\beta})}{\alpha^2})$ fresh samples from $\hat{D}$

$\bar{w} = \arg\min_{w\in L} \sum_{j=T+1}^{T+N} \mathbb{I}\{\text{sign}(\langle w, x^{(j)}\rangle) \neq y^{(j)}\}$.

Return $\bar{w}$

Theorem 11 *Let $\mathcal{P}$ be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that its marginal distribution $\mathcal{P}_x$ on $\mathbb{R}^d$ satisfies that (U, r)-anti-anti-concentration and U-anti-concentration with $U, r = \Theta(1)$, and $\|x\|_2 \leq R = O(1)$ for $x \sim \mathcal{P}_x$. Then for any $\alpha, \beta, \epsilon, \delta \in (0, 1)$, Algorithm 1 is a computationally efficient $(\epsilon, \delta, \alpha, \beta)$-NLDP Learner with sample complexity $m = O(\frac{d}{\alpha^4})$ for public unlabeled data and $n = \widetilde{O}\left(d \log^2(\frac{1}{\beta}) Poly(\log \frac{1}{\delta}, \frac{1}{\epsilon})\right)$ for private data, where the Big-$\widetilde{O}$ omits other logarithmic terms.*

Remark 12 *Firstly, we can see that the sample complexity of private data is independent of the error α. This is due to that we only need the private data to construct a Massart noise oracle with $\lambda = O(1)$. Moreover, the classifier $\hat{f}$ is a Massart noise oracle for any distribution as long as $\|x_i\|_2 \leq R$ and the assumption of anti-concentration and anti-anti-concentration is only used for Algorithm 2, which indicates that the idea of our algorithm could be used to PAC learning halfspaces with other structured distributions, as long as there is an efficient PAC learning algorithm with Massart noise.*

4.1. Proof of Theorem 11

The proof of Theorem 11 requires the following two lemmas. The first lemma suggests that $\hat{f}$ is a Massart Noise example oracle with high probability and the second lemma indicates the performance guarantee of LHMN (Algorithm 2).

Lemma 13 *Under the standard setting, for $\beta \in (0, 1)$, setting $k = O(\log(\frac{1}{\beta}))$ in Algorithm 1. Then with sample size $n \geq \widetilde{\Omega}(kd \log(\frac{1}{\beta}) Poly(\log \frac{1}{\delta}, \frac{1}{\epsilon}))$, we have the following with probability at least $1 - \beta$,*

$$\underset{(x,y)\sim\mathcal{P}}{Pr}[\hat{f}(x) \neq y] \leq \frac{3}{16}.$$

Lemma 13 suggests that for any $(x, y) \sim \mathcal{P}$, with probability no more than $\frac{3}{16}$, $\hat{f}(x)$ is adversary and returns the wrong label $\hat{f}(x) = -y$ while with probability at least $\frac{13}{16}$, it returns the correct label $\hat{f}(x) = y$. So, $\hat{f}(x)$ is in fact a Massart noise example oracle with $\lambda = \frac{3}{16}$. Before that we recall the definition of bounded distribution in Diakonikolas et al. (2020).

Definition 14 (Bounded Distribution Diakonikolas et al. (2020)) *Fix $U, R > 0$ and $t : (0, 1) \mapsto \mathbb{R}_+$. An isotropic (*i.e., *zero mean and identity covariance) distribution $\mathcal{P}_x$ on $\mathcal{R}^d$ is called $(U, R, t(\cdot))$-bounded if for any projection $(\mathcal{P}_x)_V$ of $\mathcal{P}_x$ onto a 2-dimensional subspace V, the corresponding pdf γ_V on $\mathbb{R}^d$ satisfies (U, R)-anti-anti-concentration, U-anti-concentration and for any $\alpha \in (0, 1)$, $Pr_{x\sim\gamma_V}(\|x\|_2 \geq t(\alpha)) \leq \alpha$.*

Note that since we assume $\|x\|_2 \leq R$. Thus, we always have $t(\alpha) = R$. That is, under the assumption in Theorem 11. The marginal distribution $\mathcal{P}_x$ is (U, r, R)-bounded. The next lemma about the performance guarantee of LHMN (Algorithm 2) for (U, r, R)-bounded distributions follows directly from Theorem 4.1 in Diakonikolas et al. (2020) by substituting Massart noise rate with $\lambda = \frac{3}{16}$.

Lemma 15 *Let $\mathcal{P}$ be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that the marginal distribution $\mathcal{P}_x$ on $\mathbb{R}^d$ is (U, r, R)-bounded. Let $\lambda = \frac{3}{16}$ be the upper bound on Massart noise rate. Algorithm 2 draws $m = O((\frac{U}{r})^{12} \cdot R^8 \cdot \frac{d}{\alpha^4})$ examples labeled by Massart noise example oracle and outputs a hypothesis $\bar{w}$ that satisfies $err_{\mathcal{P}}(h_{\bar{w}}) \leq \alpha$ with probability at least $1 - \beta$.*

With the above lemmas, the proof of Theorem 11 is straight forward.

Proof [Proof of Theorem 11] According to Lemma 13 and the definition of Massart noise example oracle, with probability at least $1-\beta$, $\hat{D} = \{(q_i, \hat{f}_i)\}_{i=1}^m$ can be seen as the data returned by a Massart noise example oracle with $\lambda = \frac{3}{16}$. Then, applying lemma 15, it follows directly that $err_{\mathcal{P}}(h_{\hat{w}}) \leq \alpha$ with probability at least $1 - \beta - \beta = 1 - 2\beta$. ■

5. NLDP Algorithm via Self-supervised Learning

In the previous section, we showed that if the marginal distribution $\mathcal{P}_x$ satisfies some mild assumptions, there is an NLDP algorithm using $\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon}))$ private data and $O(\frac{d}{\alpha^4})$ public unlabeled data to achieve an error of α. However, as we mentioned earlier, for smooth Generalized Linear Models with Gaussian data, there is an NLDP algorithm with sample complexity of only $O(\frac{d}{\alpha^2})$ for public data Wang et al. (2021). Thus, our question is, can we further reduce the sample complexity of public data (for other structured distributions)? In this section, we will focus on a class of distributions namely mixture distribution, which is proposed by Frei et al. (2021). We develop an (ϵ, δ)-NLDP algorithm that achieves an arbitrary classification error α using only $\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon}))$ private data and $O(\frac{d}{\alpha^2})$ public unlabeled data. We begin by introducing the mixture distribution model in Frei et al. (2021).

Algorithm 3 NLDP for Mixture distributions

1: **Input:** Private data $D = \{(x_i, y_i)\}_{i=1}^n$ with each $x_i \in \mathbb{R}^d$ satisfying $||x_i||_2 \leq R$ and $y_i \in \{\pm 1\}$; Unlabeled public data $D' = \{q_i\}_{i=1}^m$; private parameters ϵ, δ; oracle access to Logistic Loss-NLDP $\mathcal{T}_{priv}$ (Algorithm 6); error bound α; failure probability β; failure probability β; privacy parameters ϵ, δ; a constant upper bounded of $\|\mu\|_2$, ρ, where μ is the mean of the distribution of x; parameters r, U about distribution of x.

2: Run $\mathcal{T}_{priv}(C_{err} \log 2/2, R, \rho, \epsilon, \delta, D)$ and denote its output as w^{priv}, where $C_{err} = \frac{r^2}{144U}$.

3: Invoke $\{w^{(t)}\}_{t=0}^T$ =STWN($\{q_i\}_{i=1}^{T\times B}, w^{priv}$), where $B = O\left(\frac{\log(\frac{1}{\beta})}{\alpha}\right), T = \widetilde{O}\left(\frac{d(\log(\frac{1}{\beta}))^2}{\alpha}\right)$.

4: Return $\{w^{(t)}\}_{t=0}^T$

Informally, a mixture distribution model is an isotropic model generating data $(x, y) \in \mathbb{R}^d \times \{\pm 1\}$ as follows: for labels $y \in \{\pm 1\}$ and mean parameter $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{x}|y$ (conditioned on y) is a random variable with mean $y\boldsymbol{\mu}$ and identity covariance matrix. Additionally, mixture distribution model requires that $\boldsymbol{x} - y\boldsymbol{\mu}$ to satisfy anti-anti-concentration, anti-concentration (1-dim) and k-sub-exponential properties. Note that we have already introduced the definitions of anti-anti-concentration and anti-concentration (2-dim) in Definition 10. The definition of anti-concentration (1-dim) is almost the same as anti-concentration (2-dim) given in definition 10, except substituting the subspace V to a 1-dimensional subspace, which declares that the distribution assigns bounded probability mass along one-dimensional projections.

Definition 16 (U-anti-concentration (1-dim)) *Fix $U > 0$, we say an isotropic distribution $\mathcal{P}_x$ on $\mathbb{R}^d$ satisfies U-anti-concentration (1-dim) if for any projection $(\mathcal{P}_x)_V$ of $\mathcal{P}_x$ into a 1 dimensional subspace V and all $x \in V$, it holds that $\gamma_V(x) \leq U$, where γ_V the probability density function on $\mathbb{R}$.*

Definition 17 (K-sub-exponential distributions Frei et al. (2021)) *We say a distribution $\mathcal{P}_x$ is K-sub-exponential if every $x \sim \mathcal{P}_x$ is a sub-exponential random vector with sub-exponential norm at most K. In particular, if for any $\boldsymbol{v}$ with $||\boldsymbol{v}|| = 1$, $\Pr_{x\sim\mathcal{P}_x}[|\langle \boldsymbol{v}, x\rangle| \geq t] \leq e^{-\frac{t}{K}}$, then we say $\mathcal{P}_x$ is K-sub-exponential.*

Now we formally define the mixture distribution model considered in this section.

Definition 18 (Mixture distribution Frei et al. (2021)) *Let $\boldsymbol{\mu} \in \mathbb{R}^d$. Let $y = 1$ with the probability $\frac{1}{2}$ and $y = -1$ with probability $\frac{1}{2}$, and we generate $x|y \sim z + y\boldsymbol{\mu}$, where z is an isotropic K-sub-exponential distribution satisfying (U, r)-anti-anti-concentration and the U-anti-concentration (1-dim), then we say $(x, y) \sim \mathcal{P}$ is a mixture distribution with mean $\boldsymbol{\mu}$ and parameters $K, U, r = \Theta(1)$.*

Log-concave isotropic distributions such as the standard Gaussian are K-sub-exponential and satisfy U-anti-concentration (1-dim) as well as (U, r)-anti-anti-concentration (2-dim) with $K, U, r = \Theta(1)$ Frei et al. (2021). Thus, the above mixture distribution is a natural generalization of the Gaussian mixture model and can accommodate a broader class of distributions.

Similar to our previous algorithm which is based on the Massart noise model, the main idea of our NLDP algorithm for mixture distribution also consists of two parts.

(1) We first use an (ϵ, δ)-NLDP algorithm named Logistic Loss-NLDP (Algorithm 6), which is proposed by Zheng et al. (2017)), to get a private estimator w^{priv} which could achieve the error at most $C_{err} \log 2/2$ for the expected excess population risk with logistic loss by using $\widetilde{O}(d\text{Poly}(\frac{1}{\epsilon}, \log\frac{1}{\delta}))$ private data, *i.e.,*

$$\mathbb{E}[L(w^{priv}, D)] - \min_{\|w\|_2 \leq \|\boldsymbol{\mu}\|_2} \mathbb{E}[L(w, D)] \leq \frac{C_{err} \log 2}{2},$$

where $C_{err} = \frac{r^2}{144U} > 0, U, r$ are parameters of the mixture distribution, and $L(w, D) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\ell(y\langle w, x\rangle)$ with $\ell(z) = \log(1 + e^{-z})$. Based on this result, we show that w^{priv} could be thought as a pseudo labeler which achieves a sufficiently small but constant classification error at most C_{err}.

Remark 19 *The intuition of using logistic loss is that logistic loss is closely connected to 0-1 loss. Generally, logistic loss could be considered as a surrogate function of 0-1 loss. Moreover, under PAC halfspace learning setting, for any model w, its classification error could be bounded by a constant times the population risk of its logistic loss.*

(2) With the pseudo labeler, next, we use a self-training algorithm STWN in Frei et al. (2021) (Algorithm 4) to convert the weak learner (pseudo labeler) to a strong learner. The self-training algorithm can ensure that, for data coming from an isotropic mixture distribution and if there is an initial pseudo labeler w_{pl} that has small classification error, then the algorithms yield a classifier with classification error arbitrarily close to the optimal one using only unlabeled examples. In each iteration of the STWN algorithm, we first use the pseudo labeler to label a batch of unlabeled data. Then we use the gradient descent with loss function $\widetilde{\ell}$ on the pseudo labeled data to update the pseudo labeler. Note that the loss functions used in this self-training algorithm have to be "well-behaved", which is defined as follows:

Definition 20 (Well behaved loss function Frei et al. (2021)) *If the loss $\ell(z)$ is 1-Lipschitz, decreasing on the interval $[0, \infty)$ and for some constant $C_\ell \geq 1$, $\ell'(z) \geq \frac{1}{C_\ell}e^{-z}$ holds when $z > 0$, then we say the loss function is well behaved.*

Many loss functions are well behaved. For example, the exponential loss $\widetilde{\ell}(z) = e^{-z}$ and the logistic loss $\widetilde{\ell}(z) = \log(1 + e^{-z})$ satisfies the above "well behaved" definition with $C_\ell = 1$ and 2 respectively. In this paper, we will use the logistic function.

Algorithm 4 Self-training using weight normalization: STWN$(\{q_i\}_{i=1}^{T\times B}, w_{pl})$

Input: The designed estimation error α; parameters about the distribution: K, U, r; failure probability β; temperature $\sigma > 0$, batch size B and iteration T; $T \times B$ unlabeled public data $\{q_i\}_{i=1}^{T\times B}$; pseudo labeler w_{pl}.
Set step size $\eta = \widetilde{\Theta}\left(\frac{\alpha}{d(\log(\frac{1}{\beta}))^2}\right)$
Let $w^{(0)} = \frac{w_{pl}}{||w_{pl}||_2}$
for $t = 0, \cdots, T-1$ **do**
  for $i = 1 \cdots, B$ **do**
    Generate pseudo labels $\hat{y}_{B\times t+i} = \text{sign}(\langle q_i, w^{(t)}\rangle)$
  end for
  $v^{(t+1)} = w^{(t)} - \frac{\eta}{B}\sum_{i=t\times B+1}^{B\times(t+1)} \nabla\widetilde{\ell}\left(\frac{\hat{y}_i\cdot\langle q_i, w^{(t)}\rangle}{\sigma}\right)$
  $w^{(t+1)} = \frac{v^{(t+1)}}{||v^{(t+1)}||}$
end for
Return $\{w^{(t)}\}_{t=0}^{T}$

The whole picture of our NLDP algorithm for mixture distributions is given in Algorithm 3, and its theoretical guarantee is provided by the following theorem:

Theorem 21 *Assume that $(x, y) \sim \mathcal{P}$ follows a mixture distribution with $||\boldsymbol{\mu}||_2 = \Theta(1)$ and known parameters $K, U, r = \Theta(1)$, and $\|x\|_2 \leq R = O(1)$ for $x \sim \mathcal{P}_x$. Then if $\|\boldsymbol{\mu}\|_2 \geq 3K\max\{\log\frac{8}{C_{err}}, 22K\}$, for any $\alpha, \beta, \epsilon, \delta \in (0, 1)$, there exist $w \in \{w^{(t)}\}_{t=0}^{T}$ which is $(\epsilon, \delta, \alpha, \beta)$-NLDP Learner with sample complexity $m = \widetilde{O}(\frac{d\log^3\frac{1}{\delta}}{\alpha^2})$ for public unlabeled data and $n = \widetilde{O}\left(dPoly(\log\frac{1}{\delta}, \frac{1}{\epsilon})\right)$ for private data, where the Big-$\widetilde{O}$ omits other logarithmic terms.*

Remark 22 *Although the general idea of Algorithm 1 and 3 are almost the same,* i.e., *use private data to build a weak learner or a pseudo labeler and use it to transform to a strong learner. There are still several critical differences. First, in Algorithm 1 we need the weak learner w^{priv} to have a constant classification error $\lambda < \frac{1}{2}$, while in Algorithm 3 we aim to make the classification error of w^{priv} be C_{err} which needs to depend on the underlying distribution. Thus, we cannot use w^{priv} in Algorithm 1 to Algorithm 3. Second, the procedure of transforming is different, while in Algorithm 3 the labeling is adaptive, Algorithm 1 is non-adaptive. Thus, the idea of Algorithm 3 is more similar to self-supervised learning and therefore needs less public data than Algorithm 1. Thirdly, while we can guarantee that the output of Algorithm 1 is an NLDP learner, we can only ensure the existence of NLDP learner among $\{w^{(t)}\}_{t=0}^{T}$ in Algorithm 3. Finding out such a learner needs an additional one round. We leave it as an open problem for improving the algorithm.*

6. Conclusion

We studied the problem of PAC learning halfspaces in the non-interactive local differential privacy model (NLDP). Previous results either have either exponential sample complexities or they need the large margin assumption of the data. Here we considered a relaxed setting where the server has access to some additional public but unlabeled data. Specifically, under different mild assumptions on the underlying data distribution, we proposed two approaches that are based on the Massart noise model and self-supervised learning and showed that it is possible to achieve sample complexities that are only linear in the dimension and polynomial in other terms for both private and public data, which significantly improve the previous results.

Acknowledgments

Di Wang was support in part by the baseline funding BAS/1/1689-01-01, funding from the CRG grand URF/1/4663-01-01, FCC/1/1976-49-01 from CBRC and funding from the AI Initiative REI/1/4811-10-01 of King Abdullah University of Science and Technology (KAUST). He was also supported by the funding of the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

References

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pages 167–190. PMLR, 2015.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Conference on Learning Theory*, pages 152–192. PMLR, 2016.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *Conference on Learning Theory*, pages 269–282. PMLR, 2019.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013.

Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *ACM Transactions on Algorithms (TALG)*, 15(4):1–40, 2019.

Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–402. IEEE, 2020.

Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

Yuval Dagan and Vitaly Feldman. Interaction is necessary for distributed learning with privacy or communication constraints. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 450–462, 2020.

Amit Daniely and Vitaly Feldman. Locally private learning without interaction requires separation. *Advances in Neural Information Processing Systems*, 32:15001–15012, 2019.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. pages 1486–1513, 2020.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222, 2003.

Spencer Frei, Difan Zou, Zixiang Chen, and Quanquan Gu. Self-training converts weak learners to strong learners in mixture models. *arXiv preprint arXiv:2106.13805*, 2021.

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

Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

Huy Lê Nguyên, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *Algorithmic Learning Theory*, pages 704–724. PMLR, 2020.

Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34 (5):2326–2366, 2006.

Joe Near. Differential privacy at scale: Uber and berkeley collaboration. In *Enigma 2018 (Enigma 2018)*, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Adam Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 58–77. IEEE, 2017.

Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. In *Proc. 32nd Annual Conference on Advances in Neural Information Processing Systems (NeurIPS 2018)*, 2018.

Di Wang, Adam Smith, and Jinhui Xu. Noninteractive locally private learning of linear models via polynomial approximations. In *Algorithmic Learning Theory*, pages 898–903. PMLR, 2019.

Di Wang, Marco Gaboardi, Adam Smith, and Jinhui Xu. Empirical risk minimization in the non-interactive local model of differential privacy. *Journal of machine learning research*, 21(200), 2020.

Di Wang, Huangyu Zhang, Marco Gaboardi, and Jinhui Xu. Estimating smooth glm in non-interactive local differential privacy model with public unlabeled data. In *Algorithmic Learning Theory*, pages 1207–1213. PMLR, 2021.

Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR, 2017.

Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at once, use effectively: Making non-interactive locally private learning possible. In *International Conference on Machine Learning*, pages 4130–4139. PMLR, 2017.