Replicable Learning of Large-Margin Halfspaces*

Alkis Kalavasis†1, Amin Karbasi‡1,2, Kasper Green Larsen§3,
Grigoris Velegkas¶1, and Felix Zhou‖1

[1]Yale University
[2]Google Research
[3]Aarhus University

Abstract

We provide efficient replicable algorithms for the problem of learning large-margin halfspaces. Our results improve upon the algorithms provided by Impagliazzo, Lei, Pitassi, and Sorrell [STOC, 2022]. We design the first dimension-independent replicable algorithms for this task which runs in polynomial time, is proper, and has strictly improved sample complexity compared to the one achieved by Impagliazzo et al. [2022] with respect to all the relevant parameters. Moreover, our first algorithm has sample complexity that is optimal with respect to the accuracy parameter ϵ. We also design an SGD-based replicable algorithm that, in some parameters' regimes, achieves better sample and time complexity than our first algorithm.

Departing from the requirement of polynomial time algorithms, using the DP-to-Replicability reduction of Bun, Gaboardi, Hopkins, Impagliazzo, Lei, Pitassi, Sorrell, and Sivakumar [STOC, 2023], we show how to obtain a replicable algorithm for large-margin halfspaces with improved sample complexity with respect to the margin parameter τ, but running time doubly exponential in $1/\tau^2$ and worse sample complexity dependence on ϵ than one of our previous algorithms. We then design an improved algorithm with better sample complexity than all three of our previous algorithms and running time exponential in $1/\tau^2$.

1 Introduction

The replicability crisis is omnipresent in many scientific disciplines including biology, medicine, chemistry, and, importantly, AI [Baker, 2016, Pineau et al., 2019]. A recent article that appeared in Nature [Ball, 2023] explains how the reproducibility crisis in AI has a cascading effect across many other scientific areas due to its widespread applications in other fields such as medicine. Thus, an urgent goal is to design a formal framework through which we can argue about the replicability of experiments in ML. Such a theoretical framework was proposed in a recent work by Impagliazzo et al. [2022] and has been studied extensively in several learning settings [Esfandiari et al., 2023b,a, Bun et al., 2023, Kalavasis et al., 2023, Chase et al., 2023b, Dixon et al., 2023, Chase et al., 2023a, Eaton et al., 2023, Karbasi et al., 2023].

Definition 1.1 (Replicability [Impagliazzo et al., 2022]). *Let $\mathcal{R}$ be a distribution over random binary strings. A learning algorithm $\mathcal{A}$ is n-sample ρ-replicable if for any distribution $\mathcal{D}$ over inputs and two independent datasets $S, S' \sim \mathcal{D}^n$, it holds that $\mathbf{Pr}_{S,S'\sim\mathcal{D}^n, r\sim\mathcal{R}}[\mathcal{A}(S,r) \neq \mathcal{A}(S',r)] \leq \rho$.*

*Authors listed in alphabetical order.

†alvertos.kalavasis@yale.edu
‡amin.karbasi@yale.edu
§larsen@cs.au.d
¶grigoris.velegkas@yale.edu
‖felix.zhou@yale.edu

In words, this definition requires that when an algorithm $\mathcal{A}$ is executed twice on different i.i.d. datasets S, S' but using shared *internal* randomness, then the output of the algorithm is *exactly* the same, with high probability. We note that sharing the randomness across the executions is crucial in achieving this replicability guarantee. Importantly, Dixon et al. [2023] showed that without sharing randomness, it is impossible to achieve such a strong notion of replicability even for simple tasks such as mean estimation. In practice, this can be achieved by simply publishing the random seed that the ML algorithms are executed with. As we extensively discuss in Section 1.2, Definition 1.1 turns out to be connected with other notions of stability such as differential privacy and perfect generalization [Ghazi et al., 2021, Bun et al., 2023].

In this work, we study the fundamental problem of learning large-margin halfspaces, which means that no example is allowed to lie too close to the separating hyperplane. This task is related to foundational ML techniques such as the Perceptron algorithm [Rosenblatt, 1958], SVMs [Cortes and Vapnik, 1995], and AdaBoost [Freund and Schapire, 1997]. Let us recall the concept class of interest.

Definition 1.2 (Large-Margin Halfspaces). *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1, 1\}$ whose support does not contain $x = 0$. We say that $\mathcal{D}$ has linear margin τ if there exists a unit vector $w \in \mathbb{R}^d$ such that for any $(x, y) \in \mathrm{supp}(\mathcal{D})$ it holds that $y(w^\top x/\|x\|) \geq \tau$.*[1]

Following the PAC learning definition of Valiant [1984], we say that an algorithm learns with accuracy ϵ and confidence δ the class of τ-margin halfspaces in d dimensions using $n = n(\epsilon, \delta, d, \tau)$ samples and runtime $T = T(\epsilon, \delta, d, \tau)$ if, given n i.i.d. samples from any distribution $\mathcal{D}$ satisfying Definition 1.2, the algorithm outputs, in time T, a classifier $h : \mathbb{R}^d \to \{-1, 1\}$ such that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \leq \epsilon$, with probability at least $1 - \delta$.

We are interested in *replicably* learning large-margin halfspaces, i.e., designing algorithms for large-margin halfspaces that further satisfy Definition 1.1. We remark that when the feature domain is infinite, there is no replicable learning algorithm for learning halfspaces in general. Thus making some assumptions like the large-margin condition is *necessary.* In particular, Bun et al. [2023], Kalavasis et al. [2023] show that finiteness of the Littlestone dimension is a necessary condition for learnability by replicable algorithms, and it is known that even one-dimensional halfspaces over $[0, 1]$ have infinite Littlestone dimension. See Table 1.1 for a comparison of prior work and our contributions.

The work of Impagliazzo et al. [2022] provided the first replicable algorithms for τ-margin halfspaces over $\mathbb{R}^d$. The first algorithm of Impagliazzo et al. [2022], which uses the "foams" discretization scheme [Kindler et al., 2012], is ρ-replicable and returns a hypothesis h that, with probability at least $1 - \rho$, satisfies $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \leq \epsilon$. The sample complexity of this algorithm is roughly $\widetilde{O}((d\epsilon^{-3}\tau^{-8}\rho^{-2})^{1.01})$ and the runtime is exponential in d and polynomial in $1/\epsilon, 1/\rho$ and $1/\tau$. The second algorithm of Impagliazzo et al. [2022], which uses the "box" discretization scheme, is ρ-replicable and returns a hypothesis h that, with probability at least $1 - \rho$, satisfies $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \leq \epsilon$ with sample complexity $\widetilde{O}((d^3\epsilon^{-4}\tau^{-10}\rho^{-2})^{1.01})$ and runtime which is polynomial in $d, 1/\epsilon, 1/\rho$ and $1/\tau$. These two algorithms appear in the first two rows of Table 1.1.

Some remarks are in order. First, let us mention that the sample complexity of learning large-margin halfspaces in the absence of the replicability requirement is $\widetilde{O}(1/\epsilon\tau^2)$. Notice that the sample complexity of both algorithms of Impagliazzo et al. [2022] depends on the dimension d of the problem. This is unexpected since the sample complexity of non-replicable algorithms for this task is dimension-independent. In the case of the replicable algorithms of Impagliazzo et al. [2022], the dependence on the dimension appears due to a rounding/discretization step, which is crucial in establishing the replicability guarantees. Second, both algorithms of Impagliazzo et al. [2022] are *improper* in the sense that the hypothesis h they output does not correspond to a halfspace. This is due to the use of a replicable boosting routine that outputs the majority vote over multiple halfspaces. As a general note, both of these algorithms are fairly complicated: they require multiple discretization/rounding steps, and then they output a *weak* learner, which finally needs to be boosted using multiple rounds of a replicable boosting scheme. As a result, the sample complexity of their algorithms incurs a significant blow-up in the parameters ϵ, τ compared to the non-replicable setting.

[1] When we do not specify the norm, we assume the ℓ_2-norm.

Table 1.1: A comparison of prior work and our work. We denote by d the dimension, ϵ the accuracy, ρ the replicability, and τ the margin of the halfspace. We omit the logarithmic factors on the sample complexity and the runtime.

Replicable Algorithms for Large-Margin Halfspaces			
Algorithms	Sample Complexity	Running Time	Proper
Prior Work			
[ILPS22] with foams rounding	$(d\epsilon^{-3}\tau^{-8}\rho^{-2})^{1.01}$	$2^d \cdot \mathrm{poly}(1/\epsilon, 1/\rho, 1/\tau)$	No
[ILPS22] with box rounding	$(d^3\epsilon^{-4}\tau^{-10}\rho^{-2})^{1.01}$	$\mathrm{poly}(d, 1/\epsilon, 1/\rho, 1/\tau)$	No
Our Work			
Algorithm 1 (Theorem 1.3)	$\epsilon^{-1}\tau^{-7}\rho^{-2}$	$\mathrm{poly}(d, 1/\epsilon, 1/\rho, 1/\tau)$	Yes
Algorithm 2 (Theorem 1.4)	$\epsilon^{-2}\tau^{-6}\rho^{-2}$	$\mathrm{poly}(d, 1/\epsilon, 1/\rho, 1/\tau)$	Yes
[LNUZ20] via DP-to-Replicability reduction [BGH+23] (Proposition 1.5)	$\epsilon^{-2}\tau^{-4}\rho^{-2}$	$\mathrm{poly}(d) \cdot \exp\left((1/\tau)^{\frac{\log(1/(\epsilon\rho\delta))}{\tau^2}}\right)$	Yes
Algorithm 3 (Theorem 1.6)	$\epsilon^{-1}\tau^{-4}\rho^{-2}$	$\mathrm{poly}(d) \cdot \mathrm{poly}(1/\epsilon, 1/\rho, 1/\tau, 1/\delta)^{1/\tau^2}$	Yes

1.1 Our Contribution

Impagliazzo et al. [2022] leave as an open question whether their bounds are tight. In this work, we show that these bounds are sub-optimal. We provide new replicable algorithms for learning large-margin halfspaces that improve upon the results of Impagliazzo et al. [2022] in various aspects. Our algorithms have no sample dependence on d, strictly improve on the dependence on $1/\epsilon, 1/\rho$ and $1/\tau$, and are proper, meaning that they output linear models. Moreover, our Algorithm 1 and Algorithm 2 are computationally efficient while Algorithm 3 forsakes computational efficiency to achieve further improvements in sample complexity.

We now state our first algorithmic result, the proof of which can be found in Section 3.

Theorem 1.3 (Efficient Replicable Algorithm 1)**.** *Fix $\epsilon, \tau\rho, \delta \in (0, 1)$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1, 1\}$ that has linear margin τ as in Definition 1.2. There is an algorithm that is ρ-replicable and, given $m = \widetilde{O}(\epsilon^{-1}\tau^{-7}\rho^{-2}\log(1/\delta))$ i.i.d. samples $(x, y) \sim \mathcal{D}$, computes in time $\mathrm{poly}(d, 1/\epsilon, 1/\tau, 1/\rho, \log(1/\delta))$ a unit vector $w \in \mathbb{R}^d$ such that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[\mathrm{sgn}(w^\top x) \neq y] \leq \epsilon$ with probability at least $1 - \delta$.*

Algorithm 1 improves on the sample complexity of the two algorithms appearing in Impagliazzo et al. [2022], runs in polynomial time, and is proper. Our techniques follow a different path from that of Impagliazzo et al. [2022]. As we alluded to before, their approach is fairly complicated and is based on the design of a replicable weak halfspace learning algorithm and then a replicable boosting algorithm that combines multiple weak learners. Our approach is single-shot and *significantly simpler*: Consider B independent non-overlapping batches of training examples. From each batch $i \in [B]$, we find a hyperplane with normal vector $w_i \in \mathbb{R}^d$ that has $\Omega(\tau)$ margin on the training data. This can be achieved by running the standard Support Vector Machine (SVM) algorithm [Cortes and Vapnik, 1995, Vapnik, 2006]. We then aggregate our vectors to a single average normal vector $z = (1/B)\sum_{i\in[B]} w_i$. Finally, we project the vector z onto a lower-dimensional space, whose dimension does not depend on d, and we replicably round z using a rounding scheme due to Alon and Klartag [2017] for which we perform a novel analysis in the shared randomness setting. We emphasize that our algorithm gives a halfspace with the desired accuracy guarantee without the need to use any boosting schemes.

At a technical level, we avoid the dependence on the dimension d thanks to data-oblivious dimensionality reduction techniques (cf. Appendix A.3), a standard tool in the literature of large-margin halfspaces. Instead of rounding our estimates in the d-dimensional space, we first project them to a lower-dimensional space, whose dimension does not depend on d, and we perform the rounding in that space. The crucial idea is that one can use the data-obliviousness of Johnson-Lindenstrauss (JL) matrices so that the projection matrices in two distinct executions are the same since the internal randomness is shared. Another technical aspect of our algorithm that differentiates it from prior works on the design of replicable algorithms is the use of a different rounding scheme (cf. Section 2). Consider the simple case of 1-dimensional data. In the same spirit as in Impagliazzo et al. [2022], we consider a random grid. But rather than rounding the point to a fixed element of each cell of the grid (e.g., its center), we randomly round it to one of the two endpoints of the cell using shared internal randomness so that, in expectation, the rounded point is the same as the original one. This is helpful as it preserves inner products in expectation, and therefore gives better concentration properties across multiple roundings. We believe that this rounding scheme can find more applications in the replicable learning literature. The detailed proof of Theorem 1.3 can be found in Section 3.

Despite the simplicity of our algorithm, there are technical subtleties that complicate its analysis. For instance, the projection to the low-dimensional space introduces a subtle complication we need to handle. In particular, using ideas from Grønlund et al. [2020] we can show that the aggregated vector in the *high-dimensional* space has the desired generalization properties. However, when we project it to the low-dimensional space there are vectors that are now misclassified, due to the error introduced by the JL mapping. Using the guarantees of the JL projection, we show that, uniformly over the data-generating distributions, this happens for only a small fraction of the population.

We now proceed with our second algorithmic result, whose proof can be found in Section 4.

Theorem 1.4 (Efficient Replicable Algorithm 2)**.** *Fix $\epsilon, \tau, \rho, \delta \in (0,1)$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1,1\}$ that has linear margin τ as in Definition 1.2. There is an algorithm that is ρ-replicable and, given $m = \widetilde{O}(\epsilon^{-2}\tau^{-6}\rho^{-2}\log(1/\delta))$ i.i.d. samples $(x,y) \sim \mathcal{D}$, computes in time $\mathrm{poly}(d, 1/\epsilon, 1/\tau, 1/\rho, \log(1/\delta))$ a unit vector $w \in \mathbb{R}^d$ such that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[\mathrm{sgn}(w^\top x) \neq y] \leq \epsilon$ with probability at least $1-\delta$.*

Compared to Algorithm 1, our Algorithm 2 achieves better dependence on τ by incurring an additional $1/\epsilon$ factor in the sample complexity. At a technical level, as in Lê Nguyen et al. [2020], we provide a convex surrogate that upper bounds the loss $\mathbb{1}\{y(x^\top w) \leq \tau/2\}$. Running SGD on this convex surrogate provides a unit vector w that, in expectation over the data, achieves a margin of at least $\tau/2$ for an $O(\epsilon)$-mass of the population. We then apply a standard boosting trick to turn this guarantee into a high probability bound. Next, we work as in Algorithm 1: we run the above procedure B times to get $w_1, ..., w_B$ and aggregate our vectors into a single vector $z = (1/B)\sum_{i\in[B]} w_i$. Lastly, we perform a JL-projection on z and then round using the Alon-Klartag rounding scheme, as in Algorithm 1.

Computationally Inefficient Reductions from DP. It is a corollary of the works of Bun et al. [2023] and Kalavasis et al. [2023] that one can use existing differentially private (DP) algorithms in order to obtain replicable learners. In particular, following the reduction of Bun et al. [2023], one can obtain a replicable algorithm for large-margin halfspaces with better sample complexity in terms of τ, but in a *computationally inefficient* way. The idea is to take an off-the-shelf DP algorithm (recall Definition B.1) for this problem (e.g. Lê Nguyen et al. [2020]) and transform it into a replicable one. We remark that this transformation holds when the algorithm outputs *finitely* many different solutions and its running time is exponential in the number of these solutions. Fortunately, the pure DP algorithm from Lê Nguyen et al. [2020] satisfies this finite co-domain property. The formal statement of the result we get by combining these two algorithms is presented below.

Proposition 1.5 (Inefficient Replicable Algorithm; follows from Lê Nguyen et al. [2020], Bun et al. [2023])**.** *Fix $\epsilon, \tau, \rho, \delta \in (0,1)$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1,1\}$ that has linear margin τ as in Definition 1.2. There is an algorithm that is ρ-replicable and, given $m = \widetilde{O}(\epsilon^{-2}\tau^{-4}\rho^{-2}\log(1/\delta))$ i.i.d. samples $(x,y) \sim \mathcal{D}$, computes, in time $\exp\left((1/\tau)^{\frac{\log(1/(\epsilon\rho\delta))}{\tau^2}}\right)\cdot\mathrm{poly}(d)$, a unit vector $w \in \mathbb{R}^d$ such that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[\mathrm{sgn}(w^\top x) \neq y] \leq \epsilon$, with probability at least $1-\delta$.*

As mentioned above, the proof of this result follows by combining the DP-to-Replicability transformation of Bun et al. [2023] (cf. Proposition B.3) with the pure DP algorithm for learning large-margin halfspaces due to Lê Nguyen et al. [2020] (cf. Proposition B.2). We note that since the algorithm of Lê Nguyen et al. [2020] is proper and the reduction of Bun et al. [2023] is based on sub-sampling, the output of Proposition 1.5 is also a linear classifier. The main issue with this approach is that, apart from not being a polynomial time algorithm, the reduction requires a *quadratic blow-up* in the sample complexity of the provided DP algorithm.

To be more specific, the DP algorithm of Lê Nguyen et al. [2020] has sample complexity $\widetilde{O}(\epsilon^{-1}\tau^{-2})$ for accuracy ϵ and margin τ. This means that the replicable algorithm of Proposition 1.5 incurs a quadratic blow-up in the sample complexity on the parameters ϵ, τ. The cost of this transformation is tight under standard cryptographic hardness assumptions [Bun et al., 2023]. Thus, it is unlikely that we can reduce the dependence on the error parameter ϵ using such a generic transformation. We emphasize that our efficient replicable algorithm (cf. Algorithm 1) has linear sample complexity dependence on $1/\epsilon$. The blow-up on the running time of the algorithm is due to the use of *correlated sampling* in the transformation of Bun et al. [2023] which requires exponential running time in the size of the output space. We remark that in the case of the algorithm of Lê Nguyen et al. [2020], the size of the output space is already exponential in $1/\tau^2$.

In our work, we also revisit this inefficient algorithm and improve on its sample complexity and runtime, as follows:

Theorem 1.6 (Improved Inefficient Replicable Algorithm 3)**.** *Fix $\epsilon, \tau, \rho, \delta \in (0,1)$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1,1\}$ that has linear margin τ as in Definition 1.2. Then there is a ρ-replicable algorithm such that given $m = \widetilde{O}(\epsilon^{-1}\tau^{-4}\rho^{-2}\log(1/\delta))$ i.i.d. samples $(x,y) \sim \mathcal{D}$, computes in time* $\mathrm{poly}(d) \cdot \mathrm{poly}(1/\epsilon, 1/\tau, 1/\rho, 1/\delta)^{1/\tau^2}$*, a unit vector $w \in \mathbb{R}^d$ satisfying $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[\mathrm{sgn}(w^\top x) \neq y] \leq \epsilon$ with probability at least $1-\delta$.*

Compared to the DP-to-Replicability reduction, Theorem 1.6 has better dependence on $1/\epsilon$ and better running time. For the proof of this result, we refer to Section 5.

1.2 Related Work

Replicability. Pioneered by Impagliazzo et al. [2022], there has been a growing interest from the learning theory community in studying replicability as an algorithmic property in various learning tasks. Among other things, their work showed that the fundamental class of statistical queries, which appears in various settings (see e.g., Blum et al. [2003], Gupta et al. [2011], Goel et al. [2020], Fotakis et al. [2021] and the references therein) can be made replicable. Subsequently, Esfandiari et al. [2023a,b] studied replicable algorithms in the context of multi-armed bandits and clustering. Later, Eaton et al. [2023], Karbasi et al. [2023] studied replicability in the context of Reinforcement Learning. Recently, Bun et al. [2023], Kalavasis et al. [2023] established equivalences between replicability and other notions of algorithmic stability such as differential privacy (DP), and Moran et al. [2023] derived more fine-grained characterizations of these equivalences. It is worth mentioning that Malliaris and Moran [2022] had already established equivalences between various notions of algorithmic stability and finiteness of the Littlestone dimension of the underlying concept class. Inspired by Impagliazzo et al. [2022], a related line of work [Chase et al., 2023b, Dixon et al., 2023, Chase et al., 2023a] proposed and studied alternative notions of replicability such as list-replicability, where the requirement is that when the algorithm is executed multiple times on i.i.d. datasets, then the number of different solutions it will output across these executions is small.

Large-Margin Halfspaces. The problem of learning large-margin halfspaces has been extensively studied and has inspired various fundamental algorithms [Rosenblatt, 1958, Vapnik, 1999, Freund and Schapire, 1997, 1998]. In the DP setting, Blum et al. [2005] gave a dimension-dependent construction based on a private version of the perceptron algorithm. This was later improved by Lê Nguyen et al. [2020] who gave new DP algorithms for this task with dimension-independent guarantees based on the Johnson-Lindenstrauss transformation. Next, Bun et al. [2020] constructed noise-tolerant and private PAC learners for large-margin halfspaces whose sample complexity also does not depend on the dimension. Beimel et al. [2019]

and Kaplan et al. [2020] designed private algorithms for learning halfspaces without margin guarantees when the domain is finite. Bassily et al. [2022b] stated an open problem of finding optimal DP algorithms for learning large-margin halfspaces both with respect to their running time and their sample complexity. Bassily et al. [2022a] studied DP algorithms for various learning tasks with margin, including halfspaces, kernels, and neural networks. In the area of robust statistics, Diakonikolas et al. [2023] showed a statistical-computational tradeoff in the problem of PAC learning large-margin halfspaces with random classification noise. For further results on robustly learning large-margin halfspaces, we refer to Diakonikolas et al. [2019] and the references therein.

2 The Main Tool: The Alon-Klartag Rounding Scheme

Inspired by Alon and Klartag [2017], we introduce and use the following rounding scheme $\texttt{AKround}(z, \beta)$ for a point z with parameter β: let $o = (o_1, \ldots, o_k) \sim_{i.i.d.} U[0, \beta]$ be uniformly random offsets and implicitly discretize $\mathbb{R}^k$ using a grid of side length β centered at o. Let $o(z) \in \mathbb{R}^k$ denote the "bottom-left" corner of the cube in which z lies, i.e., the point obtained by rounding down all the coordinates of z. For a vector z, we let $z[i]$ be its i-th coordinate. Define $p(z)[i] \in [0, 1]$ to be such that

$$p(z)[i] \cdot o(z)[i] + (1 - p(z)[i]) \cdot (o(z)[i] + \beta) = z[i]\,.$$

Given offsets o and thresholds $u = (u_1, \ldots, u_k)$ with $u_i \sim U[0, 1]$, round a vector z to $f_{o,u}(z)$ where the i-th coordinate is equal to $o(z)[i]$ if $u_i \le p(z)[i]$ and $o(z)[i] + \beta$ otherwise. Crucially, in expectation, the rounded point $f_{o,u}(z)$ coincides with z.

The next lemma is useful in order to derive the replicability guarantees of our rounding scheme.

Lemma 2.1 (Stability of Rounding). *Let $z, z' \in \mathbb{R}^k$. Then for independent uniform offsets $o_1, \ldots, o_k \in [0, \beta]$ and thresholds $u_1, \ldots, u_k \in [0, 1]$, we have*

$$\mathop{\mathbf{Pr}}_{o,u}[f_{o,u}(z) \neq f_{o,u}(z')] \le 2\beta^{-1}\|z - z'\|_1.$$

Our novel analysis of the stability of the rounding scheme under shared randomness (cf. Lemma 2.1) demonstrates its useful properties for designing replicable algorithms. We believe these properties may be of interest beyond the scope of this work and can find applications in designing replicable algorithms for different problems.

Proof of Lemma 2.1. Fix a coordinate $i \in [k]$. The probability that $o(z)[i] \neq o(z')[i]$ is at most $|z[i] - z'[i]|\beta^{-1}$. Assume $o(z)[i] = o(z')[i]$. Then, by the definition of $p(z), p(z')$ we have that

$$\begin{aligned} |z[i] - z'[i]| &= |(p(z)[i] - p(z')[i])o(z)[i] \\ &\qquad + (p(z')[i] - p(z)[i])(o(z)[i] + \beta)| \\ &= |(p(z')[i] - p(z)[i])\beta|. \end{aligned}$$

Note then the probability of $f_{o,u}(z)[i] \neq f_{o,u}(z')[i]$ is $|p(z')[i] - p(z)[i]| = |z[i] - z'[i]|\beta^{-1}$.

By the uniform choice of u_i, we thus conclude that $\mathbf{Pr}_{o,u}[f_{o,u}(z)[i] \neq f_{o,u}(z')[i]] \le 2\beta^{-1}|z[i] - z'[i]|$. A union bound over all k coordinates implies $\mathbf{Pr}_{o,u}[f_{o,u}(z) \neq f_{o,u}(z')] \le 2\beta^{-1}\|z - z'\|_1$. ☐

Next, we show that the Alon-Klartag rounding scheme additively preserves inner products with high probability. This is formalized below.

Lemma 2.2 (Rounding preserves Inner Products). *Let $z, x \in \mathbb{R}^k$ be such that $\|x\| \le 1$. For uniform offsets $o_1, \ldots, o_k \in [0, \beta]$ and thresholds $u_1, \ldots, u_k \in [0, 1]$, we have* $\mathbf{Pr}_{o,u}\left[|f_{o,u}(z)^\top x - z^\top x| > \alpha\right] \le 2\exp\left(-2\alpha^2\beta^{-2}\right)$.

Proof of Lemma 2.2. Each of the random variables $f_{o,u}(z)[i]$ lies in an interval of length β, are independent, and have expectation $z[i]$. By linearity, $f_{o,u}(z)^\top x - z^\top x = (f_{o,u}(z) - z)^\top x$. Let $v = f_{o,u}(z) - z$. Then $\mathbf{E}[x[i] \cdot v[i]] = 0$ and $x[i] \cdot v[i]$ lies in a range of length $\beta x[i]$. By Hoeffding's inequality, we have $\mathbf{Pr}[|v^\top x| > \alpha] < 2\exp\bigl(-2\alpha^2/(\sum_i \beta^2 x[i]^2)\bigr) \leq 2\exp\bigl(-2\alpha^2\beta^{-2}\bigr)$. □

It is worth mentioning that the Alon-Klartag rounding scheme, along with dimensionality reduction techniques, was also used by Grønlund et al. [2020] in order to prove generalization bounds for SVMs.

3 Replicably Learning Large-Margin Halfspaces: Algorithm 1

In this section, we describe our first algorithm and prove its guarantees as stated in Theorem 1.3. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1, 1\}$ with linear margin $\tau \in (0, 1)$ (cf. Definition 1.2). Thus, there is a unit vector $w^\star \in \mathbb{R}^d$ such that for all $(x, y) \in \operatorname{supp}(D), x \neq 0$, we have $y(x^\top w^\star/\|x\|) \geq \tau$. Given $\epsilon, \rho, \delta \in (0, 1)$, our goal is to design a ρ-replicable learning algorithm that draws $m = m(\epsilon, \tau, \rho, \delta)$ i.i.d. samples from $\mathcal{D}$ and outputs $\hat{w} \in \mathbb{R}^d$ such that, with probability at least $1 - \delta$ over the randomness of the samples and (potentially) the internal randomness of the algorithm, it holds that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y(\hat{w}^\top x) \leq 0] \leq \epsilon$.

Description of Algorithm 1. We consider B batches of n samples each. Hence, in total, we draw nB i.i.d. samples from $\mathcal{D}$. On each batch $i \in [B]$, we run the standard SVM algorithm (cf. Lemma A.1) to find a hyperplane with normal vector $w_i \in \mathbb{R}^d$ that has margin at least $\tau/2$ on all training data in the batch. We then compute the average normal vector $z = (1/B)\sum_{i\in[B]} w_i$. Finally, we round z as described in Section 2.

Algorithm 1 Replicable Large-Margin Halfspaces

$k \leftarrow C_1\tau^{-2}\log(1/\epsilon\tau\rho\delta)$
$B \leftarrow C_2\tau^{-4}\rho^{-2}\log(1/\epsilon\tau\rho\delta)$
$n \leftarrow C_3\epsilon^{-1}\tau^{-3}\log(1/\epsilon\tau\rho\delta)$
$\beta \leftarrow C_4\tau/\log(1/\epsilon\tau\rho\delta)$
for $i = 1, 2, ...B$ **do**
  S_i = batch of n i.i.d. samples from $\mathcal{D}$
  $w_i \leftarrow \texttt{SVM}(S_i, \tau/2)$
end for
$z \leftarrow (1/B)\sum_{i\in[B]} w_i$
Draw $A \in \mathbb{R}^{k\times d}$ with $A_{i,j} \sim \mathcal{N}(0, 1/k)$ using shared randomness
$b \leftarrow \texttt{AKround}(Az, \beta)$ (cf. Section 2)
return $\hat{w} = A^\top b/\|A^\top b\|$

Correctness of Algorithm 1. A straightforward adaptation of the results of Grønlund et al. [2020] (cf. Lemma A.1) shows that, with probability $1 - \delta/(10B)$ over the samples, the classifier w_i has margin at least $\tau/4$ in a $\left(1 - O\left(\frac{\log n + \log(B/\delta)}{\tau^2 n}\right)\right)$ fraction of the population, i.e.,

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(w_i^\top x)/\|x\| < \tau/4] \leq O\left(\frac{\log n + \log(B/\delta)}{\tau^2 n}\right).$$

We denote the complement of this event as E_i and condition on not observing $\cup_{i\in[B]}E_i$.

Now under this event, for all the vectors $w_i, i \in [B]$ and all points $(x, y) \in \mathbb{R}^d \times \{-1, 1\}$, it holds that $y(w_i^\top x)/\|x\| \geq -1$, since w_i is a unit vector. Furthermore, for a $(1 - \widetilde{O}(1/\tau^2 n))$-fraction of the population, the margin is at least $\tau/4$, i.e., $y(w_i^\top x)/\|x\| \geq \tau/4$. Intuitively, this means that the vector $z = \frac{1}{B}\sum_{i\in[B]} w_i$ should have margin at least $\tau/8$, except for an $\widetilde{O}(1/(\tau^3 n))$ fraction of the population. Formally, for $(x, y) \sim \mathcal{D}$, let

Z_i be the indicator variable such that $y \cdot (w_i^\top x)/\|x\| < \tau/4$ and $Z := \sum_{i \in [B]} Z_i$. Then

$$\begin{aligned} y(z^\top x)/\|x\| &= y \left(\frac{1}{B} \sum_{i \in [B]} w_i \right)^\top (x/\|x\|) \\ &= \frac{1}{B} \sum_{i \in [B]} y \cdot w_i^\top (x/\|x\|) \\ &\geq \frac{1}{B} \left(-Z + (B - Z)\tau/4 \right) \\ &= \tau/4 - \frac{Z}{B}(1 + \tau/4) \,. \end{aligned}$$

This means that if $y(z^\top x)/\|x\| < \tau/8$, then

$$\tau/4 - \frac{Z}{B}(1 + \tau/4) < \tau/8 \implies Z > \frac{\tau}{16} \cdot B \,.$$

It suffices to bound the probability of the event that $Z > \Omega(\tau B)$ to bound the population error of z.

Notice that the summation of the fractions of the population where the w_i have margin less than $\tau/4$ is at most $O(B(\log n + \log({}^{B}\!/\!_{\delta}))/(\tau^2 n))$. As noted above, at least $\Omega(\tau B)$ of the classifiers must simultaneously have margin less than $\tau/4$ for z to misclassify x. Thus the fraction of the population where z has margin smaller than $\tau/8$ is at most

$$\begin{aligned} &O\left(\frac{B(\log n + \log({}^{B}\!/\!_{\delta}))/(\tau^2 n)}{\tau \cdot B} \right) \\ &= O\left(\frac{\log n + \log({}^{B}\!/\!_{\delta})}{\tau^3 n} \right) . \end{aligned}$$

Thus, choosing $n \geq \widetilde{\Omega}(\log({}^{B}\!/\!_{\delta})/(\tau^3 \epsilon))$ ensures that the intermediary normal vector $z = (1/B) \sum_{i=1}^{B} w_i$ satisfies $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y(z^\top x/\|x\|) < \tau/8] \leq \epsilon/10$ with probability at least $1 - \delta/10$.

The following lemma ensures that projecting and rounding in the lower dimension approximately preserves the performance of z with respect to the 0-1 loss (as opposed to the $\tau/8$-loss).

Lemma 3.1. *Fix $\epsilon, \tau, \delta \in (0, 1)$ and let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ that admits a linear classifier with τ-margin. Suppose z is a random unit vector satisfying*

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(z^\top x/\|x\|) < \tau/2] \leq \epsilon$$

with probability at least $1 - \delta$ over the draw of z. Define $k = \Omega(\tau^{-2} \log({}^{1}\!/\!_{\epsilon\delta}))$ and $\beta = O(\tau/\log({}^{1}\!/\!_{\epsilon\delta}))$. If $A \in \mathbb{R}^{k \times d}$ is a JL-matrix (cf. Appendix A.3) and $b = \texttt{AKround}(Az, \beta)$ (cf. Section 2), then $\hat{w} = A^\top b/\|A^\top b\|$ satisfies

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(\hat{w}^\top x/\|x\|) \leq 0] \leq 2\epsilon$$

with probability at least $1 - 2\delta$.

Before we prove Lemma 3.1, we note that an application of it with $\tau' = \tau/4$, $\epsilon' = \epsilon/10$, and $\delta' = \delta/10$ yields that the final output $\hat{w}$ of Algorithm 1 has 0-1 population error of at most $\epsilon/5$ with probability at least $1 - \delta/5$ as desired.

Proof. Let $\mathcal{D}_z := \mathcal{D} \mid \{y(z^\top x/\|x\|) \geq \tau/2\}$ be the distribution obtained from $\mathcal{D}$ by conditioning on z having large margin and let $\mathcal{D}_b$ denote the distribution $\mathcal{D}$ conditioned on the complement. We can decompose $\mathcal{D} = (1 - p_b) \cdot \mathcal{D}_z + p_b \cdot \mathcal{D}_b$ for $p_b \leq \epsilon$. To complete the correctness argument, we need to show that with high probability, the projection step of z, x onto the low-dimensional space Az, Ax and the rounding $b = \texttt{AKround}(Az, \beta)$ approximately preserves the inner product $y(b^\top Ax/\|x\|)$ for a $1-\epsilon$ fraction of $\mathcal{D}_z$. Then the final classifier $A^\top b$ still has low population error. In particular, it suffices to show that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}_z}[y(b^\top Ax/\|x\|) < \tau/4] \le \epsilon$ with probability at least $1-\delta$ over the random choice of A, b. Then, the total population error is at most $\epsilon + p_b \le 2\epsilon$ with probability at least $1-2\delta$.

Fix $(x, y) \in \operatorname{supp}(\mathcal{D}_z)$ and remark that

$$
\begin{aligned}
&y \cdot b^\top \frac{Ax}{\|x\|} \\
&= y \cdot (Az)^\top A\left(\frac{x}{\|x\|}\right) - y \cdot (Ax - b)^\top \frac{Ax}{\|Ax\|}.
\end{aligned}
$$

By the choices of

$$
k \ge \Omega(\tau^{-2}\log(1/\epsilon\delta)), \qquad \beta \le O(\tau/\log(1/\epsilon\delta)),
$$

the JL lemma (cf. Corollary A.5) and Alon-Kartag rounding scheme (cf. Lemma 2.2) ensures that

$$
\mathop{\mathbf{E}}_{A,b}\left[\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}_z}[y(b^\top Ax/\|x\|) < \tau/4]\right] \le \epsilon\delta.
$$

An application of Markov's inequality yields

$$
\mathop{\mathbf{Pr}}_{A,b}\left[\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}_z}[y(b^\top Ax/\|x\|) < \tau/4] > \epsilon\right] \le \delta.
$$

All in all, with probability at least $1-\delta$ over A, b, the population error is at most

$$
\begin{aligned}
&\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(A^\top b)^\top x \le 0] \\
&\le \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(b^\top Ax/\|x\|) < \Theta(\tau)] < 2\epsilon,
\end{aligned}
$$

concluding the correctness argument of our algorithm. □

Replicability of Algorithm 1. We now prove the replicability guarantees of our algorithm.

Lemma 3.2. *Fix $\epsilon, \tau, \rho, \delta \in (0,1)$. Suppose $w_1, \dots, w_B$ and $w_1', \dots, w_B'$ are i.i.d. random unit vectors for $B = \Omega(\tau^{-4}\rho^{-2}\log(1/\epsilon\tau\rho\delta))$ and $z = (1/B)\sum_{i\in[B]} w_i$, $z' = (1/B)\sum_{i\in[B]} w_i'$ are their averages. Define $k = \Theta(\tau^{-2}\log(1/\epsilon\tau\rho\delta))$ and $\beta = \Theta(\tau/\log(1/\epsilon\tau\rho\delta)$. If $A \in \mathbb{R}^{k\times d}$ is a JL-matrix (cf. Appendix A.3) and $b = \texttt{AKround}(Az, \beta)$, $b' = \texttt{AKround}(Az', \beta)$ (cf. Section 2), then $b = b'$ with probability at least $1-\rho$ over the draw of the w_i', w_i's, A, and* AKround.

Before proving Lemma 3.2, note that the w_i's are i.i.d. unit vectors across all batches and two independent executions since the samples in the $2B$ batches in the two executions are drawn from the same distribution $\mathcal{D}$ and the output of the SVM algorithm depends only on its input sample. Hence an application of Lemma 3.2 ensures that Algorithm 1 is indeed ρ-replicable.

Proof. An application of the vector Bernstein concentration inequality (cf. Lemma A.2) yields the following: let $S_1 = \left\{w_i^{(1)}\right\}_{i\in[B]}$ and $S_2 = \left\{w_i^{(2)}\right\}_{i\in[B]}$ be two sets of independent and identically distributed random vectors in $\mathbb{R}^d$ with $\left\|w_i^{(j)}\right\| \le 1$ for all $i \in [B], j \in \{1,2\}$. Then we can set $X_i = w_i^{(1)} - \mathbf{E}\, w$. Notice that $\|X_i\| \le 2$ and $\|X_i\|^2 \le 4$. Hence, an application of Lemma A.2 yields for any $t \in (0,2)$

$$
\begin{aligned}
&\mathop{\mathbf{Pr}}_{S_1}\left[\left\|\frac{1}{B}\sum w_i^{(1)} - \mathbf{E}[w]\right\| \ge t\right] \\
&\le \exp\left(-\frac{t^2 B}{32} + \frac{1}{4}\right) = \exp\bigl(\Omega(-t^2 B)\bigr)
\end{aligned}
$$

Let $z = \frac{1}{B}\sum_{i\in[B]} w_i^{(1)}, z' = \frac{1}{B}\sum_{i\in[B]} w_i^{(2)}$. Choosing $B \geq \Omega(t^{-2}\log(1/\rho))$ ensures that $\mathbf{Pr}_{S_1,S_2}[\|z-z'\| \geq t] \leq \rho/10$.

We condition on the event that $\|z - z'\| \leq t$. Since we are sharing the randomness across the two executions, we use the same JL projection matrix A. Thus our choice of $k \geq \Omega(\tau^{-2}\log(1/\rho))$ gives that $\|Az - Az'\| \leq (1+\tau)t \leq 2t$, with probability at least $1-\rho/10$ (cf. Lemma A.4). It remains to show that after the rounding step, with probability at least $1-\rho/10$, the two rounded vectors will be the same. The size of our rounding grid is $\beta = \tilde{\Theta}(\tau)$ and the target dimension of our JL-matrix is $k = \tilde{\Theta}(\tau^{-2})$. Thus by Lemma 2.1, the probability that the two points round to different vectors is $\Theta(\beta^{-1})\|Az - Az'\|_1 \leq \Theta(\beta^{-1}\sqrt{k})\|Az - Az'\|_2 = \Theta(t\sqrt{k}/\beta)$ (cf. Lemma 2.1). Thus, if we pick

$$
\begin{aligned}
&t \leq \rho\beta/\sqrt{k} \\
&B \geq \tilde{\Omega}(k/(\beta^2\rho^2)) = \Omega(\log(1/\epsilon\tau\rho\delta)/(\tau^4\rho^2)), \qquad (3.1)
\end{aligned}
$$

we complete the argument. □

Sample Complexity & Running Time of Algorithm 1. The sample complexity is $nB = \tilde{O}(\epsilon^{-1}\tau^{-7}\rho^{-2}\log(1/\delta))$. By inspection, we see that the total running time of Algorithm 1 is $\mathrm{poly}(d,n) = \mathrm{poly}(d, 1/\epsilon, 1/\tau, 1/\rho, \log(1/\delta))$.

4 Replicably Learning Large-Margin Halfspaces: Algorithm 2

Let B_1^d denote the unit ℓ_2-ball in d-dimensions. Our approach is inspired by the work of Lê Nguyen et al. [2020] that designed a similar SGD approach for learning large-margin halfspaces under differential privacy constraints. Consider the following surrogate loss $h : B_1^d \times B_1^d \times \{-1,+1\} \to \mathbb{R}_{\geq 0}$

$$h(w;x,y) := \max\left(0, 2 - \frac{2}{\tau}y(x^\top w)\right) \geq \mathbb{1}\{y(x^\top w) < \tau/2\}.$$

We remark that $h(w;x,y) \geq 1$ when $y(x^\top w) \leq \tau/2$ and $h(w;x,y) = 0$ when $y(x^\top w) \geq \tau$. Also, since $x, w \in B_1^d$, an application of the Cauchy-Schwartz inequality reveals that $h(w;x,y) \in [0, 2+2/\tau]$. Finally, h is piecewise linear with each piece being $2/\tau$-Lipschitz. Hence, h is $O(1/\tau)$-Lipschitz.

Description of Algorithm 2. Fix $\epsilon \in (0,1)$. We seek to minimize the following loss function over the ball B_1^d:

$$f_{\mathcal{D}}(w) := \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[h(w;x,y)] + \frac{\epsilon}{10}\|w\|^2.$$

By construction, the co-domain of $f_{\mathcal{D}}$ lies within $[0, 2+2/\tau+\epsilon/10] \subseteq [0, O(1/\tau)]$. Note also that $f_{\mathcal{D}}$ is an upper bound on the $\tau/2$-population loss, i.e.,

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(x^\top w) < \tau/2] \leq f(w)\,.$$

First, regarding the minima of $f_{\mathcal{D}}$, note that any vector $w \in B_1^d$ achieving a margin of τ satisfies $f_{\mathcal{D}}(w) \leq \epsilon/10$. This is because $h(w;x,y) = 0$ for all $(x,y) \in \mathrm{supp}(\mathcal{D})$. As a result, $\min_{w\in B_1^d} f_{\mathcal{D}}(w) \leq \epsilon/10$. Second, let us consider an $\epsilon/10$-optimal solution w' with respect to $f_{\mathcal{D}}$, i.e.,

$$f_{\mathcal{D}}(w') - \min_{w\in B_1^d} f \leq \epsilon/10\,.$$

The above discussion implies that $f_{\mathcal{D}}(w') \leq \epsilon/5$ and is thus a $\tau/2$-margin classifier for an $\epsilon/5$-fraction of the population, i.e., w' satisfies

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(x^\top w') < \tau/2] \leq \epsilon/5\,.$$

Note that we may assume without loss of generality that the marginal of $\mathcal{D}$ over features is supported over B_1^d since we normalize the input $x \mapsto x/\|x\|$ before applying a classifier w.

Since f is $O(\epsilon + 1/\tau) = O(1/\tau)$-Lipschitz and $\Omega(\epsilon)$-strongly convex, we can apply the following standard result:

Theorem 4.1 (Theorem 6.2 in Bubeck et al. [2015])**.** *Let f be μ-strongly convex with minimizer w_* and assume that the (sub)gradient oracle $g(w)$ satisfies $\mathbf{E}[\|g(w)\|^2] \leq G^2$. Then after T iterations, projected stochastic gradient descent* `SGD`$(\mathcal{D}, f, T)$ *with step size $\eta_t = {}^2/_{\mu(t+1)}$ satisfies*

$$\mathbf{E}\left[f\left(\sum_{t=1}^{T} \frac{2t}{T(T+1)} w_t\right)\right] - f(w_*) \leq \frac{2G^2}{\mu(T+1)}.$$

Since $G^2 = O(1/\tau^2)$, $\mu = \Omega(\epsilon)$ and $f(w_*) \leq \epsilon/10$, choosing $T \geq \Omega(\epsilon^{-2}\tau^{-2})$ yields an $\epsilon/10$-optimal solution in expectation. Repeating this process independently for a small number of times and outputting the one with the lowest objective yields an $\epsilon/5$-optimal solution with high probability.

Lemma 4.2. *Let B_1^d be the unit ℓ_2-ball in d dimensions. Fix $\epsilon, \tau, \delta \in (0, 1)$ and let $\mathcal{D}$ be a distribution over $B_1^d \times \{\pm 1\}$ that admits a linear τ-margin classifier. There is an algorithm* `boostSGD`$(\mathcal{D}, \epsilon, \tau, \delta)$ *that outputs a unit vector $\tilde{w} \in \mathbb{R}^d$ such that $f_{\mathcal{D}}(\tilde{w}) \leq \min_{w \in B_1^d} f_{\mathcal{D}}(w) + \epsilon$ with probability at least $1 - \delta$. Moreover,* `boostSGD` *has sample complexity $\tilde{O}(\epsilon^{-2}\tau^{-2}\log(1/\delta))$ and running time $\mathrm{poly}(1/\epsilon, 1/\tau, \log(1/\delta), d)$.*

Proof. Let $T = \Omega(\tau^{-2}\epsilon^{-2})$. Run `SGD`$(\mathcal{D}, T)$ for $n = O(\log(1/\delta))$ times to obtain solutions $w_1, \dots, w_n$. Theorem 4.1 ensures that we attain an $\epsilon/10$-optimal solution in expectation for each $w_i, i \in [n]$. Markov's inequality then guarantees that we attain an $\epsilon/5$-optimal solution with probability at least 1/2 in each repetition. But then with probability at least $1 - \delta/10$, at least one of the n solutions is $\epsilon/5$-optimal.

By a Hoeffding bound, we can estimate each $f_{\mathcal{D}}(w_i) \in [0, O(1/\tau)]$ up to an additive $\epsilon/10$ error with probability at least $1 - \delta/10$ using $O(n\epsilon^{-2}\tau^{-2}\log(n/\delta))$ samples. Outputting the classifier with the lowest estimated objective yields a $3\epsilon/10$-optimal solution with probability at least $1 - \delta/5$.

Normalizing the selected classifier to a unit vector can only decrease h since the feasible region is B_1^d and incurs an additional loss of $\epsilon/10$ due to the regularizer. Thus we have a $4\epsilon/10$-optimal solution with probability at least $1 - \delta/5$.

The total sample complexity is $O(nT) + O(n\epsilon^{-2}\tau^{-2}\log(n/\delta)) = \tilde{O}(\epsilon^{-2}\tau^{-2}\log(1/\delta))$. □

Next, we repeat `boostSGD` and take an average to ensure concentration before proceeding as in Algorithm 1 with the random projection and rounding in the lower dimensional space. Compared to Algorithm 1, we obtain an improved sample dependence on τ for Algorithm 2 since taking an average of ϵ-optimal solutions to a convex objective function yields an ϵ-optimal solution. However, we pay an extra factor of ϵ in order to run `SGD` as a subroutine.

Algorithm 2 Replicable Large-Margin Halfspaces

$n = C_1\epsilon^{-2}\tau^{-2}\log(1/\epsilon\tau\rho\delta)$
$B \leftarrow C_2\tau^{-4}\rho^{-2}\log(1/\epsilon\tau\rho\delta)$
$k \leftarrow C_3\tau^{-2}\log(1/\epsilon\tau\rho\delta)$
$\beta \leftarrow C_4\tau/\log(1/\epsilon\tau\rho\delta)$
for $i \leftarrow 1, \dots, B$ **do**
  $\tilde{S}_i \leftarrow n$ samples from $\mathcal{D}$
  $S_i \leftarrow \{(x/\|x\|, y) : (x, y) \in \tilde{S}_i\}$
  $w_i \leftarrow$ `boostSGD`$(S_i, \epsilon/10, \tau, \delta/B)$ (cf. Lemma 4.2)
end for
$z \leftarrow (1/B)\sum_{i \in [B]} w_i$
Draw $A \in \mathbb{R}^{k \times d}$ with $A_{i,j} \sim \mathcal{N}(0, 1/k)$ using shared randomness
$b \leftarrow$ `AKround`(Az, β) (cf. Section 2)
return $\hat{w} = A^\top b/\|A^\top b\|$

Correctness of Algorithm 2. From the choice of $n \geq \Omega(\epsilon^{-2}\tau^{-2}\log(^B/_\delta))$, Lemma 4.2 ensures that each unit vector w_i produced in Algorithm 2 is $\epsilon/10$-optimal with probability at least $1-\delta/(10B)$. Hence with probability at least $1-\delta/10$, every w_i is $\epsilon/10$-optimal with respect to $f_{\mathcal{D}}$. By Jensen's inequality, $z = (1/B)\sum_{i=1}^B w_i$ is $\epsilon/10$-optimal with probability at least $1-\delta/10$. But then by the choice of $f_{\mathcal{D}}$ as a convex surrogate loss for $\mathbb{1}\{y(x^\top w) < \tau/2\}$, z satisfies $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y(z^\top x/\|x\|) < \tau/2] \leq 2\epsilon/10$ with probability at least $1-\delta/10$. In other words, the unit vector z has a population $\tau/2$-loss of at most $2\epsilon/10$ with probability at least $1-\delta/10$. But then similar to the correctness of Algorithm 1, an application of Lemma 3.1 concludes the proof of correctness.

Replicability of Algorithm 2. Similar to the correctness of Algorithm 1, we note that the output w_i of each execution of `boostSGD` is an i.i.d. unit vector. Thus an application of Lemma 3.2 yields the replicability guarantees of Algorithm 2.

Sample Complexity & Running Time of Algorithm 2. The sample complexity is $nB = \widetilde{O}(\epsilon^{-2}\tau^{-6}\rho^{-2}\log(^1/_\delta))$ as required. Once again, we see that Algorithm 2 terminates in $\operatorname{poly}(d, 1/\epsilon, 1/\tau, 1/\rho, \log(^1/_\delta))$ by inspection.

5 Replicably Learning Large-Margin Halfspaces: Algorithm 3

In this section, we provide an algorithm whose sample complexity scales as $\widetilde{O}(\epsilon^{-1}\tau^{-4}\rho^{-2}\log(^1/_\delta))$ and has running time $\operatorname{poly}(d)\cdot(\operatorname{poly}(1/\epsilon, 1/\rho, 1/\tau, 1/\delta))^{1/\tau^2}$. We remark that the sample complexity is better than the one obtained from the DP transformation in Proposition 1.5. Moreover, the running time is exponentially better than that obtained through the DP transformation.

Before this, we state a very useful result due to Bun et al. [2023] related to the sample complexity of replicably learning finite hypothesis classes, which is used in the proof of Theorem 1.6.

Proposition 5.1 (Theorem 5.13 in Bun et al. [2023])**.** *Consider a finite concept class $\mathcal{H}$. There is a ρ-replicable agnostic PAC learner* `rLearnerFinite` *with accuracy ϵ and confidence δ for $\mathcal{H}$ with sample complexity $n = O\left(\frac{\log^2|H| + \log(^1/_{\rho\delta})}{\epsilon^2\rho^2}\log^3(^1/_\rho)\right)$. Moreover, if $\mathcal{D}$ is realizable then the sample complexity drops to $\widetilde{O}(\epsilon^{-1}\rho^{-2}\log^2|H|)$. Finally, the algorithm terminates in time* $\operatorname{poly}(|\mathcal{H}|, n)$.

Description of Algorithm 3 Let us first provide the algorithm's pseudo-code.

Algorithm 3 Replicable Large-Margin Halfspaces

- $k \leftarrow C\tau^{-2}\log(^1/_{\epsilon\tau\rho\delta})$
- $n \leftarrow C'\epsilon^{-1}\tau^{-4}\rho^{-2}\log(^1/_{\epsilon\rho\tau\delta})$
- $S \leftarrow$ batch of n i.i.d. samples from $\mathcal{D}$
- Draw $A \in \mathbb{R}^{k\times d}$ with $A_{i,j} \sim \mathcal{N}(0, 1/k)$ using the shared randomness
- $S_A \leftarrow (Ax_1, y_1), \ldots, (Ax_n, y_n)$
- $\mathcal{H}_\tau \leftarrow$ a $(\tau/20)$-net over vectors of length at most 1 in $\mathbb{R}^k$
- $b \leftarrow$ output of `rLearnerFinite` from Proposition 5.1 with input $S_A, \mathcal{H}_\tau$
- $\hat{w} \leftarrow A^\top b/\|A^\top b\|$

Similar to the other algorithm, we first start by using a JL matrix to project the training set to a k-dimensional space, for $k = \widetilde{\Theta}(\tau^{-2})$. Then, we use a $(\tau/20)$-net to cover all the unit vectors of this k-dimensional space, so the size of the net is $(^{C''}/_\tau)^{\widetilde{O}(\tau^{-2})}$, for some absolute constant $C'' > 0$. We think of these points of the net as our hypothesis class $\mathcal{H}_\tau$. By the properties of the JL transform, we can show that with high probability $1-\delta/10$, there exists a vector in this class that classifies the entire training set correctly. Moreover, we can show that this classifier has small generalization error. This is formalized in the following result, which is essentially a high-probability version of Lemma A.6 from Lê Nguyen et al. [2020].

Lemma 5.2. *Fix $\epsilon', \delta_{JL} \in (0,1)$. Let $\mathcal{D}$ satisfy Definition 1.2 with margin τ and suppose $w^\star$ satisfies $y(w^\star)^\top x \geq \tau$ for every $(x,y) \in \mathrm{supp}(\mathcal{D})$. For a JL-matrix $A \in \mathbb{R}^{k \times d}$, as stated in Lemma A.3 with $k = \Omega(\tau^{-2}\log(1/\delta_{JL}))$, let $G_A \subseteq \mathbb{R}^d \times \{-1,1\}$ be the set of points (x,y) of $\mathrm{supp}(\mathcal{D})$ that satisfy*

- $\left|\|Ax\|^2 - \|x\|^2\right| \leq \tau\|x\|^2/100$, *and,*
- $y(Aw^\star/\|Aw^\star\|)^\top(Ax/\|Ax\|) \geq 96\tau/100$.

Let E_1 be the event (over A) that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[(x,y) \in G_A] \geq 1-\epsilon'$ and E_2 be the event (over A) that $\left|\|Aw^\star\|^2 - \|w^\star\|^2\right| \leq \tau\|w^\star\|^2/100$. Then it holds that $\mathbf{Pr}_A[E_1 \cap E_2] \geq 1 - \delta_{JL}/\epsilon'$.

The proof of Lemma 5.2 appears in Appendix A.3.1. One way to view Lemma 5.2 is that, with probability $1-\delta_{JL}/\epsilon'$ over the random choice of A, the classifier $Aw^\star/\|Aw^\star\|$ will have $96\tau/100$ margin on a $1-\epsilon'$ fraction of $\mathcal{D}$, where the choice of ϵ' will be specified later according to Lemma 5.2. Let us condition on this event for the rest of the proof, which we call E_r. Let $\tilde{w}^\star$ be the point of the net that $Aw^\star/\|Aw^\star\|$ is rounded to. Recall that we round to the closest point on a $\tau/20$-net of the unit ball with respect to the ℓ_2-norm, so that $\tilde{w}^\star$ is $\tau/20$ close to the normalized version of $Aw^\star$. Notice that under the event E_r, for all points $(x,y) \in G_A$ we have that

$$
\begin{aligned}
&(\tilde{w}^\star)^\top \frac{Ax}{\|Ax\|} \\
&= \frac{(Aw^\star)^\top}{\|Aw^\star\|}\frac{Ax}{\|Ax\|} - \left(\frac{Aw^\star}{\|Aw^\star\|} - \tilde{w}^\star\right)^\top \frac{Ax}{\|Ax\|} \\
&\geq \frac{(Aw^\star)^\top}{\|Aw^\star\|}\frac{Ax}{\|Ax\|} - \left\|\frac{Aw^\star}{\|Aw^\star\|} - \tilde{w}^\star\right\| \cdot \frac{\|Ax\|}{\|Ax\|} \\
&\geq 96\tau/100 - \tau/20 > 9/(10\tau)\,,
\end{aligned}
$$

where the first inequality follows from Cauchy-Schwartz and the second inequality from the definition of the net. Since the hypothesis class $\mathcal{H}_\tau$ has finite size, we can use Proposition 5.1 from Bun et al. [2023] which states that $\widetilde{O}(\epsilon^{-1}\rho^{-2}\log(1/\delta)\log^2|\mathcal{H}|)$ samples are sufficient to ρ-replicably learn a hypothesis class $\mathcal{H}$ in the *realizable* setting with error at most ϵ. One technical complication we need to handle is that $\mathcal{D}$ is not necessarily realizable with respect to $\mathcal{H}_\tau$. Nevertheless, under E_r, we have shown that for $\tilde{w}^\star \in \mathcal{H}_\tau$ it holds that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y((\tilde{w}^\star)^\top Ax/\|Ax\|) < 9\tau/10] \leq \epsilon'$. Let us denote by $\mathcal{D}_r$ the distribution $\mathcal{D}$ conditioned on the event that $y((\tilde{w}^\star)^\top Ax/\|Ax\|) \geq 9\tau/10$ and $\mathcal{D}_b$ its complement, i.e., $\mathcal{D}$ conditioned on $y((\tilde{w}^\star)^\top Ax/\|Ax\|) < 9\tau/10$. Then, we can express $\mathcal{D} = (1-p_b)\cdot\mathcal{D}_r + p_b\cdot\mathcal{D}_b$, where $p_b \leq \epsilon'$. Hence, in a sample of size n from $\mathcal{D}$, with probability at least $1 - n\cdot\epsilon'$ we only see samples drawn i.i.d. from $\mathcal{D}_r$. Let us call this event E_r' and condition on it. Let us choose $\epsilon' = \delta/(10n), \delta_{JL} = \epsilon'^2/10 = \delta^2/1000n$ so that $\mathbf{Pr}[E_r \cap E_r'] \geq 1-\delta$.

Under the events E_r, E_r', we can use the replicable learner from Bun et al. [2023] to learn the *realizable* distribution $\mathcal{D}_r$. Run the algorithm from Proposition 5.1 with parameters $\delta/10, \epsilon/10, \rho/10$.[2] Since $|\mathcal{H}| = (C/\tau)^{\tau^{-2}}$ for some absolute constant C, we see that we need $n = \widetilde{O}\left(\epsilon^{-1}\tau^{-4}\rho^{-2}\log(1/\delta)\right)$ samples.

Replicability of Algorithm 3. Let us condition on the events E_r, E_r' across two executions of the algorithm. Since, the matrix A is the same, due to shared randomness, under these events the samples that the learner of Bun et al. [2023] receives are i.i.d. from the same *realizable* distribution. Then, the replicability guarantee follows immediately from the guarantees of Proposition 5.1 and the fact that all the events we have conditioned on occur with probability at least $1-\rho/2$.

[2] We assume that $\delta < \rho, \epsilon$, otherwise we normalize δ to get the bound.

Correctness of Algorithm 3. Here we examine the population error of the output of Proposition 5.1. Note that this error is analyzed with respect to the original distribution $\mathcal{D}$, which is not necessarily realizable, rather than $\mathcal{D}_r$, which is the distribution on which we run the learning algorithm.

Let us again condition on the events E_r, E_r', and the event that the output of Proposition 5.1 satisfies the generalization bound it states. We assume without loss of generality that $\delta \le \epsilon/2$, otherwise we set $\delta = \epsilon/2$ without affecting the overall sample complexity of our algorithm. Notice that these events occur with probability at least $1-\delta$. Let b be the output of the algorithm. Then we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y(b^\top x) < 0] &= (1-p_b)\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}_r}[y(b^\top x)<0] \\ &\quad + p_b \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}_b}[y(b^\top x)<0] \\ &\le (1-p_b)\epsilon/10 + p_b < \epsilon.\end{aligned}$$

This concludes the proof.

Sample Complexity & Running Time of Algorithm 3. As noted before, we need $n = \widetilde{O}\left(\epsilon^{-1}\tau^{-4}\rho^{-2}\log(1/\delta)\right)$ samples. Since we apply Proposition 5.1, we incur a running time of $\mathrm{poly}(d) \cdot \mathrm{poly}(1/\epsilon, 1/\rho, 1/\tau, 1/\delta)^{1/\tau^2}$.

6 Conclusion

In this work, we have developed new algorithms for replicably learning large-margin halfspaces. Our results vastly improve upon prior work on this problem. We believe that many immediate questions for future research arise from our work. First, it is natural to ask whether there are efficient algorithms that can achieve the $\widetilde{O}(\epsilon^{-1}\tau^{-4}\rho^{-2}\log(1/\delta))$ sample complexity bound of Algorithm 3. Also, it would be interesting to see if there are any (not necessarily efficient) replicable algorithms whose sample complexity scales as $\widetilde{O}(\epsilon^{-1}\tau^{-2}\rho^{-2})$ or if there is some inherent barrier to pushing the dependence on τ below τ^{-4}. Finally, our analysis of Algorithm 1 is pessimistic in the sense that it uses a pigeonhole principle argument to establish that the fraction of the population where the aggregate vector does not have margin $\Omega(\tau)$ is $\widetilde{O}(1/\tau^3 n)$. It would be interesting to see whether this bound can be improved to $\widetilde{O}(1/\tau^2 n)$ using a different analysis, which would reduce the overall sample complexity dependence of the algorithm on τ.

References

Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281, 2001. 17, 18

Noga Alon and Bo'az Klartag. Optimal compression of approximate inner products and dimension reduction. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650. IEEE, 2017. 3, 6

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 2016. 1

Philip Ball. Is ai leading to a reproducibility crisis in science? *Nature*, 624(7990):22–25, 2023. 1

Raef Bassily, Mehryar Mohri, and Ananda Theertha Suresh. Differentially private learning with margin guarantees. *Advances in Neural Information Processing Systems*, 35:32127–32141, 2022a. 6

Raef Bassily, Mehryar Mohri, and Ananda Theertha Suresh. Open problem: Better differentially private learning algorithms with margin guarantees. In *Conference on Learning Theory*, pages 5638–5643. PMLR, 2022b. 6

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. 5

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003. 5

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005. 5

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. 11

Mark Bun, Marco Leandro Carmosino, and Jessica Sorrell. Efficient, noise-tolerant, and private learning via boosting. In *Conference on Learning Theory*, pages 1031–1077. PMLR, 2020. 5

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Jessica Sorrell, and Satchit Sivakumar. Stability is stable: Connections between replicability, privacy, and adaptive generalization. *arXiv preprint arXiv:2303.12921*, 2023. 1, 2, 4, 5, 12, 13, 19

Zachary Chase, Bogdan Chornomaz, Shay Moran, and Amir Yehudayoff. Local borsuk-ulam, stability, and replicability. *arXiv preprint arXiv:2311.01599*, 2023a. 1, 5

Zachary Chase, Shay Moran, and Amir Yehudayoff. Replicability and stability in learning. *arXiv preprint arXiv:2304.03757*, 2023b. 1, 5

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 2, 3

Ilias Diakonikolas, Daniel Kane, and Pasin Manurangsi. Nearly tight bounds for robust proper learning of halfspaces with a margin. *Advances in Neural Information Processing Systems*, 32, 2019. 6

Ilias Diakonikolas, Jelena Diakonikolas, Daniel M Kane, Puqian Wang, and Nikos Zarifis. Information-computation tradeoffs for learning margin halfspaces with random classification noise. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2211–2239. PMLR, 2023. 6

Peter Dixon, A Pavan, Jason Vander Woude, and NV Vinodchandran. List and certificate complexities in replicable learning. *arXiv preprint arXiv:2304.02240*, 2023. 1, 2, 5

Eric Eaton, Marcel Hussing, Michael Kearns, and Jessica Sorrell. Replicable reinforcement learning. *arXiv preprint arXiv:2305.15284*, 2023. 1, 5

Hossein Esfandiari, Alkis Kalavasis, Amin Karbasi, Andreas Krause, Vahab Mirrokni, and Grigoris Velegkas. Replicable bandits. In *The Eleventh International Conference on Learning Representations*, 2023a. 1, 5

Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velegkas, and Felix Zhou. Replicable clustering. *arXiv preprint arXiv:2302.10359*, 2023b. 1, 5

Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Efficient algorithms for learning from coarse labels. In *Conference on Learning Theory*, pages 2060–2079. PMLR, 2021. 5

Casper Benjamin Freksen. An introduction to johnson-lindenstrauss transforms. *arXiv preprint arXiv:2103.00564*, 2021. 17

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 2, 5

Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 209–217, 1998. 5

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level private learning via correlated sampling. *arXiv preprint arXiv:2110.11208*, 2021. 2

Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. *Advances in Neural Information Processing Systems*, 33:2147–2158, 2020. 5

Allan Grønlund, Lior Kamma, and Kasper Green Larsen. Near-tight margin-based generalization bounds for support vector machines. In *International Conference on Machine Learning*, pages 3779–3788. PMLR, 2020. 4, 7, 17

Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 803–812, 2011. 5

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. *arXiv preprint arXiv:2201.08430*, 2022. 1, 2, 3, 4, 5

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. 17, 18

William B Johnson. Extensions of lipshitz mapping into hilbert space. In *Conference modern analysis and probability, 1984*, pages 189–206, 1984. 17, 18

Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. *arXiv preprint arXiv:2305.14311*, 2023. 1, 2, 4, 5

Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. *Advances in Neural Information Processing Systems*, 33:13976–13985, 2020. 6

Amin Karbasi, Grigoris Velegkas, Lin F Yang, and Felix Zhou. Replicability in reinforcement learning. *arXiv preprint arXiv:2305.19562*, 2023. 1, 5

Guy Kindler, Anup Rao, Ryan O'Donnell, and Avi Wigderson. Spherical cubes: optimal foams from computational hardness amplification. *Communications of the ACM*, 55(10):90–97, 2012. 2

Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1895–1904. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/kohler17a.html`. 17

Huy Lê Nguyen, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *Algorithmic Learning Theory*, pages 704–724. PMLR, 2020. 4, 5, 10, 13, 18, 19

Maryanthe Malliaris and Shay Moran. The unstable formula theorem revisited. *arXiv preprint arXiv:2212.05050*, 2022. 5

Shay Moran, Hilla Schefler, and Jonathan Shafer. The bayesian stability zoo. *arXiv preprint arXiv:2310.18428*, 2023. 5

Joelle Pineau, Koustuv Sinha, Genevieve Fried, Rosemary Nan Ke, and Hugo Larochelle. Iclr reproducibility challenge 2019. *ReScience C*, 5(2):5, 2019. 1

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 2, 5

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 2

Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 1999. 5

Vladimir Vapnik. *Estimation of dependences based on empirical data.* Springer Science & Business Media, 2006. 3

A Deferred Tools

A.1 SVM guarantees

The following result is a restatement of Theorem 2 from Grønlund et al. [2020]

Lemma A.1 (SVM Generalization Guarantee [Grønlund et al., 2020])**.** *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{-1,1\}$. Let $n \in \mathbb{N}, S = (x_1, y_1), \dots, (x_n, y_n) \sim \mathcal{D}^n$, and $w \in \mathbb{R}^d$ be a unit vector such that $y_i \left(w^\top \frac{x_i}{\|x_i\|}\right) \geq \tau, \forall i \in [n]$ Then, for every $\delta > 0$ with probability at least $1-\delta$ over the random draw of S, it holds that*

$$\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}}[y \cdot (w^\top x/\|x\|) < \tau/2] \leq O\left(\frac{\log n + \log(1/\delta)}{\tau^2 n}\right).$$

We remark that even though the result of Grønlund et al. [2020] is stated for the generalization error with respect to the misclassification probability, i.e., $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y \cdot (w^\top x/\|x\|) \leq 0]$, their argument also applies to the $\tau/2$-margin loss, i.e., $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[y \cdot (w^\top x/\|x\|) < \tau/2]$, via a straightforward modification of some constants. In more detail, the only needed change is in the proof of part 1 of Claim 10 in Grønlund et al. [2020]. Here the first condition says $y \cdot (x^\top w) \leq 0$ and could be made e.g. $y \cdot (x^\top w) \leq \tau/4$. Their result then holds with $\tau/4$ instead of $\tau/2$.

A.2 Vector-Valued Bernstein Concentration Inequality

We will use the following concentration inequality for the norm of random vectors.

Lemma A.2 (Vector Bernstein; [Kohler and Lucchi, 2017])**.** *Let $X_1, \dots, X_B$ be independent random vectors with common dimension d satisfying the following for all $i \in [B]$:*

(i) $\mathbf{E}[X_i] = 0$

(ii) $\|X_i\| \leq \mu$

(iii) $\mathbf{E}[\|X_i\|^2] \leq \sigma^2$

Let $Z := \frac{1}{B}\sum_{i=1}^B X_i$. Then for any $t \in (0, \sigma^2/\mu)$,

$$\mathbf{Pr}[\|Z\| \geq t] \leq \exp\left(-\frac{t^2 B}{8\sigma^2} + \frac{1}{4}\right).$$

A.3 Johnson-Lindenstrauss Lemma

The remarkable result of Johnson [1984] states that an appropriately scaled random orthogonal projection matrix preserves the norm of a unit vector with high probability. Indyk and Motwani [1998] showed that it suffices to independently sample each entry of the matrix from the standard normal distribution. Achlioptas [2001] further simplified the construction to independently sample each entry from the Rademacher distribution. See Freksen [2021] for a detailed survey of the development.

From hereonforth, we say that a random matrix $A \in \mathbb{R}^{k\times d}$ is a *JL-matrix* if either $A_{ij} \sim_{i.i.d.} \mathcal{N}(0, 1/k)$ or $A_{ij} \sim_{i.i.d.} U\{-1/\sqrt{k}, +1/\sqrt{k}\}$.

We first state the standard distributional formulation of JL.

Lemma A.3 (Distributional JL; [Johnson, 1984, Indyk and Motwani, 1998, Achlioptas, 2001])**.** *Fix $\epsilon, \delta_{JL} \in (0,1)$. Let $A \in \mathbb{R}^{k\times d}$ be a JL-matrix for $k = \Omega(\epsilon^{-2}\log(1/\delta_{JL}))$. Then for any $x \in \mathbb{R}^d$,*

$$\Pr_A[\left|\|Ax\|^2 - \|x\|^2\right| > \epsilon\|x\|^2] \le \delta_{JL} \,.$$

Let T be a set of vectors. By applying Lemma A.3 to the $O(|T|^2)$ vectors $u - v$ for $u, v \in T$ and taking a union bound, we immediately deduce the following result.

Lemma A.4 (JL Projection; [Johnson, 1984, Indyk and Motwani, 1998, Achlioptas, 2001])**.** *Fix $\epsilon, \delta_{JL} \in (0,1)$. Consider a set T of d-dimensional vectors and a JL-matrix $A \in \mathbb{R}^{k\times d}$ for $k = \Omega(\epsilon^{-2}\log(|T|/\delta_{JL}))$. Then,*

$$\Pr_A\left[\exists\, u, v \in T : \left|\|A(u-v)\|^2 - \|u-v\|^2\right| > \epsilon\|u-v\|^2\right] \le \delta_{JL} \,.$$

An application of Lemma A.4 towards the polarization identity for $z, x \in \mathbb{R}^d$

$$4z^\top x = \|z+x\|^2 - \|z-x\|^2$$

yields the following inner product preservation guarantee.

Corollary A.5 (JL Inner Product Preservation)**.** *Fix $\epsilon, \delta_{JL} \in (0,1)$. Let $A \in \mathbb{R}^{k\times d}$ be a JL-matrix for $k = \Omega(\epsilon^{-2}\log(1/\delta_{JL}))$. Then, for any $x, z \in \mathbb{R}^d$,*

$$\Pr_A[|z^\top x - (Az)^\top Ax| > \epsilon\|z\|\cdot\|x\|] \le \delta_{JL} \,.$$

The next lemma is another simple implication of the distributional JL.

Lemma A.6 (Lemma 5 in Lê Nguyen et al. [2020])**.** *Let $\mathcal{D}$ satisfy Definition 1.2 with margin τ. For a JL-matrix A as stated in Lemma A.3, let $G_A \subseteq \mathbb{R}^d \times \{-1,1\}$ be the set of points (x,y) of the population that satisfy*

- $\left|\|Ax\|^2 - \|x\|^2\right| \le \tau\|x\|^2/100$ *and*
- $y(Aw^\star/\|Aw^\star\|)^\top(Ax/\|Ax\|) \ge 96\tau/100.$

Then it holds that $\mathbf{Pr}_{A,(x,y)\sim\mathcal{D}}[(x,y) \in G_A] \ge 1 - \delta_{JL}$.

A.3.1 The Proof of Lemma 5.2

We finally prove Lemma 5.2, whose statement we repeat below for convenience.

Lemma 5.2. *Fix $\epsilon', \delta_{JL} \in (0,1)$. Let $\mathcal{D}$ satisfy Definition 1.2 with margin τ and suppose $w^\star$ satisfies $y(w^\star)^\top x \ge \tau$ for every $(x,y) \in \mathrm{supp}(\mathcal{D})$. For a JL-matrix $A \in \mathbb{R}^{k\times d}$, as stated in Lemma A.3 with $k = \Omega(\tau^{-2}\log(1/\delta_{JL}))$, let $G_A \subseteq \mathbb{R}^d \times \{-1,1\}$ be the set of points (x,y) of $\mathrm{supp}(\mathcal{D})$ that satisfy*

- $\left|\|Ax\|^2 - \|x\|^2\right| \le \tau\|x\|^2/100$, *and,*
- $y(Aw^\star/\|Aw^\star\|)^\top(Ax/\|Ax\|) \ge 96\tau/100.$

Let E_1 be the event (over A) that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[(x,y) \in G_A] \ge 1 - \epsilon'$ and E_2 be the event (over A) that $\left|\|Aw^\star\|^2 - \|w^\star\|^2\right| \le \tau\|w^\star\|^2/100$. Then it holds that $\mathbf{Pr}_A[E_1 \cap E_2] \ge 1 - \delta_{JL}/\epsilon'$.

Proof of Lemma 5.2. We know from Lemma A.6 that

$$\mathop{\mathbf{E}}_A\left[\Pr_{(x,y)\sim\mathcal{D}}[(x,y) \notin G_A]\right] \le \delta_{JL} \,,$$

so Markov's inequality gives that

$$\Pr_A\left[\Pr_{(x,y)\sim\mathcal{D}}[(x,y)\notin G_A]\geq\epsilon'\right]\leq\frac{\mathbf{E}_A\left[\Pr_{(x,y)\sim\mathcal{D}}[(x,y)\notin G_A]\right]}{\epsilon'}\leq\frac{\delta_{JL}}{\epsilon'}\,.$$

Similarly, the guarantees of the JL projection immediately yields

$$\Pr_A\left[\left|\|Aw^\star\|^2-\|w^\star\|^2\right|\leq\tau\|w^\star\|^2/100\right]\leq\delta_{JL}.$$

□

B Details for Proposition 1.5

B.1 Differential Privacy

For $a,b,\alpha,\beta\in[0,1]$, let $a\approx_{\alpha,\beta}b$ denote the statement $a\leq e^\alpha b+\beta$ and $b\leq e^\alpha a+\beta$. We say that two probability distributions P,Q are (α,β)-indistinguishable if $P(E)\approx_{\alpha,\beta}Q(E)$ for any measurable event E.[3]

Definition B.1 (Approximate Differential Privacy)**.** *A learning rule A is an n-sample (α,β)-differentially private if for any pair of datasets $S,S'\in(\mathcal{X}\times\{0,1\})^n$ that differ on a single example, the induced posterior distributions $A(S)$ and $A(S')$ are (α,β)-indistinguishable.*

B.2 The Results of Lê Nguyen et al. [2020] and Bun et al. [2023]

Proposition 1.5 is based on combining the following two results. The first is a differentially private learner for large-margin halfspaces from Lê Nguyen et al. [2020].

Proposition B.2 (Theorem 6 in Lê Nguyen et al. [2020])**.** *Let $\alpha,\tau,\epsilon,\delta>0$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d\times\{-1,1\}$ that has linear margin τ as in Definition 1.2. There is an algorithm that is $(\alpha,0)$-differentially private and, given $m=\widetilde{O}(\alpha^{-1}\epsilon^{-1}\tau^{-2})$ i.i.d. samples $(x,y)\sim\mathcal{D}$, computes in time $\exp\bigl(1/\tau^2\bigr)\mathrm{poly}(d,1/\alpha,1/\epsilon,\log(1/\delta))$ a normal vector $w\in\mathbb{R}^d$ such that $\Pr_{(x,y)\sim\mathcal{D}}[\mathrm{sgn}(w^\top x)\neq y]\leq\epsilon$, with probability at least $1-\delta$.*

We also use the DP-to-Replicability reduction appearing in Bun et al. [2023].

Proposition B.3 (Corollary 3.18 in Bun et al. [2023])**.** *Fix $n\in\mathbb{N}$, sufficiently small $\rho\in(0,1)$, $\epsilon,\delta\in(0,1)$ and $\alpha,\beta>0$. Let $\mathcal{A}:\mathcal{X}^n\to\mathcal{Y}$ be an n-sample (α,β)-differentially private algorithm with **finite output space** solving a statistical task with accuracy ϵ and failure probability δ. Then there is an algorithm $\mathcal{A}':\mathcal{X}^m\to\mathcal{Y}$ that is ρ-replicable and solves the same statistical task with $m=O(\rho^{-2}n^2)$ samples with accuracy ϵ and failure probability δ.*

[3] We use the notation $(\alpha,\beta)-$DP instead of the more common $(\epsilon,\delta)-$DP to be consistent with the notation of the rest of the paper regarding the accuracy, probability of failure of the learning algorithms.