PAC Learning of Halfspaces with Malicious Noise in Nearly Linear Time

Jie Shen
Stevens Institute of Technology

Abstract

We study the problem of efficient PAC learning of halfspaces in $\mathbb{R}^d$ in the presence of the malicious noise, where a fraction of the training samples are adversarially corrupted. A series of recent works have developed polynomial-time algorithms that enjoy near-optimal sample complexity and noise tolerance, yet leaving open whether a *linear-time* algorithm exists and matches these appealing statistical performance guarantees. In this work, we give an affirmative answer by developing an algorithm that runs in time $\tilde{O}(md)$, where $m = \tilde{O}(\frac{d}{\epsilon})$ is the sample size and $\epsilon \in (0,1)$ is the target error rate. Notably, the computational complexity of all prior algorithms suffer either a high order dependence on the problem size, or is implicitly proportional to $\frac{1}{\epsilon^2}$ through the sample size. Our key idea is to combine localization and an approximate version of matrix multiplicative weights update method to progressively downweight the contribution of the corrupted samples while refining the learned halfspace.

1 INTRODUCTION

We study the problem of learning homogeneous halfspaces in the probably approximately correct (PAC) model of Valiant (1984). This is one of the most extensively studied problems in machine learning, dating back to the 1950s (Rosenblatt, 1958). In the absence of noise, it is known that the problem can be solved in polynomial time using linear programming (Maass and Turán, 1994). In this work, we consider learning halfspaces in the presence of the malicious noise (Valiant, 1985), perhaps the strongest noise model.

Let $\mathcal{X} := \mathbb{R}^d$ be the instance space, $\mathcal{Y} := \{-1, 1\}$ be the label space, and $\mathcal{H} := \{w \in \mathbb{R}^d : x \mapsto \operatorname{sign}(w \cdot x), \|w\| = 1\}$ be the hypothesis class of homogeneous halfspaces. Let D be a distribution on $\mathcal{X}$. The learner is given access to a sample generation oracle $\mathrm{EX}(D, w^*)$ that works as follows:

Definition 1 (Malicious noise)**.** Each time the learner requests a sample, with probability $1 - \eta$, the oracle $\mathrm{EX}(D, w^*)$ randomly draws an instance x according to D and returns the clean sample $(x, \operatorname{sign}(w^* \cdot x))$; with probability η, the oracle may return an arbitrary pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. We call w^* the target halfspace and η the noise rate.

This is a significantly more challenging noise model than other broadly studied ones such as the adversarial noise (Haussler, 1992; Kearns et al., 1992), in the sense that the malicious oracle can corrupt both instances and labels. It is also worth noting that the oracle is assumed to have unbounded computational power and know the learning algorithm (including the internal randomness). The goal of the learner is to output a halfspace $\hat{w} \in \mathcal{H}$ such that its error $\mathrm{err}_D(w) := \Pr_{x \sim D}\big(\operatorname{sign}(\hat{w} \cdot x) \neq \operatorname{sign}(w^* \cdot x)\big) \leq \epsilon$ for any prescribed error rate $\epsilon \in (0, 1)$.

Generally speaking, there are two important dimensions along with algorithmic design: the statistical performance and the computational complexity. Most prior works in this space aimed at developing algorithms with favorable statistical performance guarantees, especially the achievability of optimal noise tolerance. For example, Kearns and Li (1988) showed that the information-theoretic limit of the noise tolerance is $\frac{\epsilon}{1+\epsilon}$ and developed an efficient algorithm with noise tolerance $\eta = \Omega(\epsilon/d)$. This was later improved to $\Omega(\epsilon/\sqrt{d})$ with smooth boosting (Servedio, 2003). Under distributional assumptions on D, the noise tolerance was improved to $\tilde{\Omega}(\epsilon/d^{1/4})$ by Kalai et al. (2005) and to $\Omega(\epsilon^2/\log(d/\epsilon))$ by using outlier removal (Klivans et al., 2009). The seminal work of Awasthi et al. (2017) settled the near-optimal noise tolerance $\eta = \Omega(\epsilon)$ by leveraging the idea of soft outlier removal into the margin-based active learning framework of Balcan et al. (2007). With an improved analysis, a very recent work of Shen (2021b) showed that the polynomial-time algorithm of Awasthi et al. (2017) essentially achieves near-optimal noise tolerance, sample complexity, and label complexity simultaneously.

In contrast to the rich set of statistically efficient algorithms, less is explored for designing practical algorithms that are scalable to large-scale problems – though this is a central theme in machine learning. In fact, as Awasthi et al. (2017);

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

Shen (2021b) suggested, their algorithm needs to apply the ellipsoid method as a subroutine, which roughly runs in $O(md^4)$ time (Bubeck, 2015) that is prohibitive when the instances lie in a high-dimensional space. Other aforementioned algorithms appear to run faster, yet comprising a significant degree of noise tolerance and sample complexity. For example, Kalai et al. (2005) showed that a simple averaging scheme runs in $O(md)$ time yet with noise tolerance $\Omega(\epsilon/\sqrt{d})$; with a preprocessing step that runs in time $O(n^2 d)$, the noise tolerance can be improved to $\tilde{\Omega}(\epsilon/d^{1/4})$. Likewise, Klivans et al. (2009) achieves better noise tolerance in terms of the dependence on d but their algorithm runs in $O(md^3/\log m)$. In a nutshell, there always exists an unpleasant tradeoff between noise tolerance and running time in prior works; see Table 1 for a summary.[1] Thus, a natural and important question remains open:

> *Does there exist an algorithm that achieves optimal noise tolerance and computational complexity simultaneously?*

1.1 Warmup: Learning via Mean Estimation

Had our goal been just obtaining nearly linear-time algorithm with good noise tolerance (better than those listed in Table 1), there would be a naive approach that makes use of a fast robust mean estimation algorithm Dong et al. (2019) as a black box and PAC learns $\mathcal{H}$ under a strong condition that D be the standard Gaussian. Roughly speaking, in robust mean estimation, there is an underlying distribution D' that has an unknown mean $\mu^* \in \mathbb{R}^d$ but the covariance matrix is known. The learner has access to instances $z_1, \dots, z_m$ drawn from D' among which $\eta < 1/2$ fraction are adversarially corrupted, and the goal is to output $\hat{\mu}$ such that $\|\hat{\mu} - \mu^*\| \leq \epsilon$. Returning to the problem setup of learning halfspaces, under the condition that D, the underlying distribution on $\mathcal{X}$, is the standard Gaussian distribution $N(0, I_{d\times d})$, it is well-known that for any clean sample (x, y) with $y = \mathrm{sign}(w^* \cdot x)$, $z := \sqrt{\frac{\pi}{2}} yx$ is an unbiased estimate of w^*. Thus, given $(x_1, y_1), \dots, (x_m, y_m)$, we can construct $z_1, \dots, z_m$ with $z_i = \sqrt{\frac{\pi}{2}} y_i x_i$, and the clean samples z_i can be thought of as being generated from a sub-gaussian distribution D' with unknown mean w^*. This motivates the following naive approach for learning halfspace with malicious noise:

> **Naive Approach**: Let $m = \Omega(d/\epsilon^2)$. Draw $\{(x_i, y_i)\}_{i=1}^m$ from $\mathrm{EX}(D, w^*)$ and construct $Z = \{z_i\}_{i=1}^m$ with $z_i = \sqrt{\frac{\pi}{2}} y_i x_i$. Run the algorithm of Dong et al. (2019) on Z which outputs $\hat{w} \in \mathbb{R}^d$.

We can then apply Theorem 2.2 of Dong et al. (2019) to show that the output $\hat{w}$ is close to w^* in ℓ_2-norm, which translates into a PAC guarantee in view of a result from Balcan and Long (2013).

Proposition 2 (Naive approach). *Assume D is $N(0, I_{d\times d})$. The naive approach satisfies the following: given any target error $\epsilon \in (0,1)$ and failure probability $\delta \in (0,1)$, if $\eta \leq c_0' \frac{\epsilon}{\log(1/\epsilon)}$ for some constant $c_0' > 0$, by drawing $m = \Omega(d/\epsilon^2)$ samples, it returns a hypothesis $\hat{w} \in \mathcal{H}$ such that with probability $1 - \delta$, $\Pr_{x\sim D}\big(\mathrm{sign}(\hat{w} \cdot x) \neq \mathrm{sign}(w^* \cdot x)\big) \leq \epsilon$. In addition, it runs in time $\tilde{O}(md)$.*

First of all, such naive approach already improves upon Kalai et al. (2005); Klivans et al. (2009) in all aspects discussed so far (see Table 1). It also significantly reduces the running time of Awasthi et al. (2017); Shen (2021b) with a little sacrifice in noise tolerance. However, there are two drawbacks in such black-box application of Dong et al. (2019) compared to the guarantees in Awasthi et al. (2017); Shen (2021b). First, the success of the naive approach requires a strong distributional condition that D is standard Gaussian (which is needed to ensure the mean of z_i equal w^* for clean samples), while the results of Awasthi et al. (2017); Shen (2021b) hold under the significantly more general condition that D be isotropic log-concave (Lovász and Vempala, 2007; Vempala, 2010). Second, the noise tolerance and the sample size above are suboptimal. In particular, the sample complexity n is proportional to $1/\epsilon^2$ while that of Shen (2021b) scales as $1/\epsilon$.

1.2 Main Results

Our main contribution is the first *nearly linear-time* algorithm with near-optimal noise tolerance and sample complexity, under the significantly weaker assumption that the distribution D is isotropic log-concave (Lovász and Vempala, 2007; Vempala, 2010).

Theorem 3 (Main result). *Assume that D is isotropic log-concave. There exists an algorithm $\mathcal{A}$ satisfying the following: given any target error $\epsilon \in (0,1)$ and failure probability $\delta \in (0,1)$, if $\eta \leq c \cdot \epsilon$ for some small constant $c > 0$, by making $m = \frac{d}{\epsilon} \cdot \mathrm{polylog}(\frac{d}{\epsilon\delta})$ calls to $\mathrm{EX}(D, w^*)$, it returns a hypothesis $\hat{w} \in \mathcal{H}$ such that with probability $1 - \delta$, $\Pr_{x\sim D}\big(\mathrm{sign}(\hat{w} \cdot x) \neq \mathrm{sign}(w^* \cdot x)\big) \leq \epsilon$. In addition, it runs in time $\tilde{O}\big(md \cdot \log^4 \frac{1}{\epsilon}\big)$.*

Remark 4. Similar to Awasthi et al. (2017), we can consider an active learning setting where upon receiving a request from the learner, the adversary $\mathrm{EX}(D, w^*)$ generates a sample (x, y) as before but only returns the instance x. The learner must make another call to a label revealing oracle EX^y to obtain the label y. We can show that the total number of calls to EX^y, i.e. the *label complexity*, is $\tilde{O}\big(d \cdot \mathrm{polylog}(\frac{1}{\epsilon})\big)$, which is near-optimal (Kulkarni et al., 1993). As the integration of active learner has been fairly standard in the literature, we leave it to interested readers.

[1] We are doing our best to locate the specific theorem of each related work in Table 1, but some bounds were not explicitly stated yet were implied by the proof. Also, the bound on m in Awasthi et al. (2017) was $\tilde{O}(d^3)$, but it has recently been improved by Shen (2021b); we are using the improved bound in the table.

Table 1: Comparison to Prior Robust Algorithms. There is always an unpleasant tradeoff between noise tolerance and running time in prior works. For example, Kalai et al. (2005) designed a linear-time algorithm, but with the worst noise tolerance, while Awasthi et al. (2017) achieved the best noise tolerance but with very high computational cost. Our work achieves the best of the two worlds (up to a very mild $\mathrm{polylog}(\frac{1}{\epsilon})$ factor in the running time).

Work	Noise tolerance	Running time	m
Theorem 12 of Kalai et al. (2005)	$\Omega(\epsilon/\sqrt{d})$	$O(md)$	$\tilde{O}(d^2/\epsilon^2)$
Theorem 4 of Kalai et al. (2005)	$\tilde{\Omega}(\epsilon/d^{1/4})$	$O(m^2 d)$	$\tilde{O}(d^2/\epsilon^2)$
Theorem 1 of Klivans et al. (2009)	$\Omega(\epsilon^2/\log(d/\epsilon))$	$O(md^3/\log m)$	$\tilde{O}(d^3/\epsilon^2)$
Theorem 1.2 of Awasthi et al. (2017) and Shen (2021b)	$\Omega(\epsilon)$	$O(md^4 \cdot \log \frac{1}{\epsilon})$	$\tilde{O}(d/\epsilon)$
This work (Theorem 3)	$\Omega(\epsilon)$	$\tilde{O}(md \cdot \log^4 \frac{1}{\epsilon})$	$\tilde{O}(d/\epsilon)$

Remark 5. In the very special regime of $d < \log \frac{1}{\epsilon}$, our runtime bound is worse than that of Awasthi et al. (2017). Such regime is less interesting in practice though.

Remark 6. It is possible to generalize the distributional condition to the family of well-behaved distributions as set out in Diakonikolas et al. (2020b), but in our case we will still need a sub-exponential tail bound since the success of outlier removal and analysis of error rate both depend on such condition.

1.3 Related Works

Learning halfspaces with noise is one of the most important problems in machine learning. When the labels are adversarially corrupted and there is no distributional assumption on D, it was shown that even weak PAC learning is computationally hard (Guruswami and Raghavendra, 2006; Feldman et al., 2006; Daniely, 2016). Under the Massart label noise condition (Sloan, 1988; Massart and Nédélec, 2006), it was shown that an error rate less than $\eta + \epsilon$ can be achieved by efficient algorithms (Diakonikolas et al., 2019a; Chen et al., 2020). On the other hand, a series of recent works showed that when the underlying instance distribution is well-behaved (e.g. isotropic log-concave), it is possible to establish efficient PAC learning algorithms with error rate less than ϵ when the samples are corrupted by various types of label noise, such as the Massart noise (Massart and Nédélec, 2006; Awasthi et al., 2015, 2016; Yan and Zhang, 2017; Zhang et al., 2020; Diakonikolas et al., 2020b; Zhang and Li, 2021), the Tsybakov noise (Tsybakov, 2004; Diakonikolas et al., 2020c,a), the adversarial/agnostic noise (Haussler, 1992; Kearns et al., 1992; Awasthi et al., 2017; Shen, 2021a; Diakonikolas et al., 2021b). This paper studies the regime where both instances and labels are adversarially corrupted, thus is much more challenging (Awasthi et al., 2017; Shen, 2021b). In addition to the works that we have discussed, there are other interesting works that considered learning of more general hypothesis classes such as polynomial threshold functions and intersections of halfspaces (Diakonikolas et al., 2018), or studied performance guarantee when the underlying hypothesis class is sparse halfspaces (Shen and Zhang, 2021). It is worth mentioning that a more general noise model termed nasty noise was coined out in Bshouty et al. (2002), where the oracle is allowed to remove clean samples. Our results can be extended to this noise model, though we do not pursue it here.

The problem of learning halfspaces with malicious noise is, conceptually, related to robust mean estimation, where the learner is given a set of instances among which a large fraction are drawn from some distribution with unknown mean and the rest are adversarially corrupted, and the goal is to approximate the mean. The problem roots in robust statistics since the 1960s (Tukey, 1960; Huber, 1964), yet only recently have efficient algorithms been established (Diakonikolas et al., 2016; Lai et al., 2016). After that, there have been a flurry of developments concerning, for example, faster implementation (Diakonikolas et al., 2017a; Cheng et al., 2019; Dong et al., 2019; Hopkins et al., 2020), improved sample complexity under structural assumptions (Balakrishnan et al., 2017; Diakonikolas et al., 2019c; Zeng and Shen, 2022), statistical-query lower bounds (Diakonikolas et al., 2017b); see Diakonikolas and Kane (2019); Diakonikolas et al. (2021a) for a comprehensive survey. We will mostly be using the results from the appealing work of Dong et al. (2019), whose primary idea is to identify multiple directions where corrupted samples may lie on to accelerate outlier removal.

1.4 Roadmap

The rest of the paper is organized as follows. In Section 2, we describe the main algorithm that achieves the guarantees announced in Theorem 3. In Section 3, we present theoretical analysis of the proposed algorithm. Section 4 concludes the paper. All the proof details are deferred to the appendix.

1.5 Notations

For a vector v, we denote its ℓ_1-norm, ℓ_2-norm, and ℓ_∞-norm by $\|v\|_1$, $\|v\|$, and $\|v\|_\infty$ respectively. For two vectors u and v, we use $u \leq v$ (or $u \geq v$) for element-wise comparison. For a matrix M, we write $\|M\|$ for its spectral norm, which is the largest singular value. Let n be a positive integer. We write $[n] := \{1, \dots, n\}$. The letters c and C, as well as their subscript variants such as c_1 and C_1, are reserved for specific absolute constants. For two quantities f and g, we write $f = O(g)$ if $f \leq K \cdot g$ for some constant $K > 0$, $f = \Omega(g)$ if $f \geq K \cdot g$, and $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$. To ease the expression, sometimes we will use the $\tilde{O}(\cdot)$ notation, where $f = \tilde{O}(g)$ means $f \leq K \cdot g \cdot \operatorname{polylog}(g)$; likewise, $f = \tilde{\Omega}(g)$ reads as $f \geq K \cdot g / \operatorname{polylog}(g)$.

2 MAIN ALGORITHMS

In this section, we present our main algorithm. It follows from the one of Awasthi et al. (2017) and our key technical contribution can be regarded as a nearly linear-time implementation with provable guarantees. In particular, we will combine the matrix multiplicative weights update method (Arora et al., 2012) into margin-based active learning (Balcan et al., 2007) to accelerate outlier removal, and will use an online gradient descent method to refine the learned halfspaces.

2.1 Margin-Based Active Learning

To design an algorithm that achieves the guarantees in Theorem 3, we will use the margin-based active learning algorithm of Awasthi et al. (2017), which exhibits state-of-the-art statistical guarantees (Shen, 2021b). We first review the general idea.

Recall that $w^* \in \mathbb{R}^d$ is the target halfspace. Suppose that we are given a halfspace u such that $\|u - w^*\| \leq r$ for some constant radius $r > 0$. For example, one can pick an arbitrary unit vector $u \in \mathbb{R}^d$ whose distance to w^* must be less than 2. It is then sensible to consider a localized hypothesis space $W := \{w \in \mathbb{R}^d : \|w - u\| \leq r\}$ as a trust region of w^* since $W \supset w^*$. An elegant result from Balcan and Long (2013) (see Theorem 21 therein) tells that as long as the instance distribution D is isotropic log-concave, for any $u' \in W$, its error rate in $\bar{X}_{u,b} := \{x \in \mathbb{R}^d : |u \cdot x| \geq b\}$ must be at most $c_0 r$ for arbitrarily small constant $c_0 > 0$, provided $b = C_0 r$ for some sufficiently large constant $C_0 > 0$. Therefore, the learner only needs to find a hypothesis $u' \in W$ that incurs small error rate in the band $X_{u,b} := \{x \in \mathbb{R}^d : |u \cdot x| \leq b\}$. In Awasthi et al. (2017), it was shown that a small constant error in $X_{u,b}$ suffices to certify an improved estimate u' in the sense that $\left\|u' - w^*\right\| \leq \frac{r}{2}$. Thus, by iterating with $O(\log \frac{1}{\epsilon})$ phases, it is possible to find a halfspace whose distance to w^* is at most ϵ, which would imply the desired error rate.

Now we delve a little into the algorithmic details. In light of the above discussion, a crucial step in each phase is to find a hypothesis $u' \in W$ with $\operatorname{err}_{D_{u,b}}(u') \leq \kappa$ for some small constant $\kappa > 0$ even in the presence of the malicious noise, where $D_{u,b}$ is the distribution D conditioned on the event $x \in X_{u,b}$. To this end, Awasthi et al. (2017) proposed to use the ellipsoid method to find a feasible solution to the following linear program for a given corrupted set $S = \{(x_i, y_i)\}_{i=1}^n \subset X_{u,b}$:

$$\begin{aligned} &\min_{q=(q_1,\dots,q_n)} \langle 0, q\rangle \\ &\text{s.t. } 0 \leq q \leq \frac{1}{n}, \quad \sum_{i=1}^n q_i \geq (1-\xi), \\ &\sup_{w \in W} \sum_{i=1}^n q_i (w \cdot x_i)^2 \leq O(b^2 + r^2), \end{aligned} \tag{1}$$

where ξ is an estimate of the fraction of corrupted samples in S (which can be shown to behave as a constant here). The weights $\{q_i\}_{i=1}^n$ should be thought of as indicating how likely the i-th sample (x_i, y_i) is a clean sample. In this sense, the ideal weights are: $q_i = \frac{1}{n}$ for all clean samples (x_i, y_i) and $q_i = 0$ for all corrupted ones. The first constraint in the above expression can be viewed as a convex relaxation to the hard constraint $q_i \in \{0, \frac{1}{n}\}$; the second constraint in Eq. (1) ensures that a roughly $1 - \xi$ fraction of samples will be assigned with weight close to $\frac{1}{n}$ – recall that the total number of clean samples in S is $(1-\xi)n$. The last constraint enforces that corrupted samples will not be assigned with large weight: the term $O(b^2 + r^2)$ is precisely an upper bound on the left hand side had there been no corrupted samples. The way that we impose such variance constraint follows from the intuition that if the oracle were to force the learner to produce a hypothesis that deviates far from w^*, it has to concentrate its power on generating samples lying roughly on some directions such that the hinge loss on these samples are large (which enforces any optimizer to find a solution that fits these corrupted losses). Such observation is due to Blum et al. (1996) to handle the random classification noise and was then utilized in Klivans et al. (2009) for learning with the malicious noise.

Equipped with the weight vector q from Eq. (1), it then minimizes a reweighted empirical hinge loss up to a constant $\kappa > 0$ to produce a new iterate:

$$\min_{w \in W} \ell_\tau(w; S, q) := \sum_{i=1}^n q_i \cdot \max\left\{0, 1 - \frac{1}{\tau} y_i w \cdot x_i\right\}, \tag{2}$$

where $\tau = \Theta(b)$ is a proper scaling factor that would be useful to control the sample complexity.

We follow this pipeline and our main technical contribution is to design nearly linear-time algorithms REWEIGHT (Algorithm 2) and OPTIMIZE (Algorithm 4) that solve (1) and (2) respectively. The main algorithm is outlined in Algorithm 1, where in Step 7 we prune away all samples thate have large ℓ_2-norm. We will see that this is crucial to establish bounded computational complexity for OPTIMIZE yet does not hurt the statistical performance since it is possible to show that

Algorithm 1 Main Algorithm

```
Require: Error rate $\epsilon \in (0,1)$, failure probability $\delta \in (0,1)$, sample generation oracle $\mathrm{EX}(D, w^*)$.
Ensure: Halfspace $\hat{w}$ with $\mathrm{err}_D(\hat{w}) \leq \epsilon$ with probability $1-\delta$.
Initialize $w_0$ as the zero vector in $\mathbb{R}^d$.
$k_{\max} \leftarrow \Theta(\log \frac{1}{\epsilon})$.
for phases $k = 1, 2, \dots, k_{\max}$ do
    $b_k \leftarrow \Theta(2^{-k})$, $r_k \leftarrow \Theta(2^{-k})$, $\tau_k \leftarrow \Theta(2^{-k})$,
    $W_k \leftarrow \{w \in \mathbb{R}^d : \|w - w_{k-1}\| \leq r_k\}$, $X_k \leftarrow \{x \in \mathbb{R}^d : |w_{k-1} \cdot x| \leq b_k\}$
    $S' \leftarrow$ request $n_k$ samples in $X_k$ from $\mathrm{EX}(D, w^*)$ by rejection sampling.
    $S \leftarrow S' \cap \big\{(x,y) : \|x\| \leq \gamma_k\big\}$ where $\gamma_k = \sqrt{d}\log\big(\frac{2en_k}{c_8 b_k \delta_k}\big)$.
    $p = (p_1, \dots, p_{n_k}) \leftarrow$ REWEIGHT$(S, \gamma_k, \delta_k/4)$, $p \leftarrow p/\|p\|_1$.
    $w_k \leftarrow$ OPTIMIZE$(S, p, W_k, b_k, \gamma_k, \delta_k/4)$, $w_k \leftarrow w_k/\|w_k\|$.
end for
return $\hat{w} \leftarrow w_{k_{\max}}$.
```

Algorithm 2 REWEIGHT

```
Require: A sample set $S = \{(x_i, y_i)\}_{i=1}^n$, scalar $\gamma$ such that $\max_{i \in [n]} \|x_i\| \leq \gamma$, a spectral norm estimate $\lambda^*$, failure probability $\delta'$.
Ensure: A weight vector $\hat{p} = (\hat{p}_1, \dots, \hat{p}_n)$ on $S$ such that $\|M(\hat{p})\| \leq 1350\lambda^*$.
$p^{(1)} \leftarrow (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$, $J \leftarrow \log_{4/3}(\frac{\gamma^2}{\lambda^*})$.
for $j = 1, \dots, J$ do
    $\lambda^{(j)} \leftarrow$ APPROXEV$\big(p^{(j)}, S, \frac{1}{10}, \frac{\delta'}{2J}\big)$.
    if $\lambda^{(j)} \leq 1500\lambda^*$ then return $\hat{p} \leftarrow p^{(j)}$.
    $p^{(j+1)} \leftarrow$ REFINE$(p^{(j)}, S, \gamma, \lambda^*, \frac{\delta'}{2J})$.
end for
return $\hat{p} \leftarrow p^{(J+1)}$.
```

Algorithm 3 REFINE$(q, S, \gamma, \lambda^*, \delta'')$

```
Ensure: A weight vector $\hat{q} \in Q_S$ such that $\|M(\hat{q})\| \leq \frac{3}{4}\|M(q)\|$ or $\|M(\hat{q})\| \leq 1350\lambda^*$.
$q^{(1)} \leftarrow q$, $T \leftarrow 8\log d$.
for $t = 1, \dots, T$ do
    $\lambda^{(t)} \leftarrow$ APPROXEV$(q^{(t)}, S, \frac{1}{10}, \frac{\delta''}{2T})$
    if $\lambda^{(t)} \leq \frac{1}{2}\lambda^{(1)}$ or $\lambda^{(t)} \leq 1200\lambda^*$ then return $\hat{q} \leftarrow q^{(t)}$.
    $\tilde{\beta}^{(t)} \leftarrow$ MMWUScore$\big(S, q^{(1)}, \dots, q^{(t)}, \frac{\delta''}{2T}\big)$.
    if $\langle q^{(t)}, \tilde{\beta}^{(t)} \rangle \leq \frac{1}{5}\lambda^{(1)}$ then
        $q^{(t+1)} \leftarrow q^{(t)}$.
    else
        $q^{(t+1)} \leftarrow$ 1D-FILTER$(q^{(t)}, \tilde{\beta}^{(t)}, \frac{1}{45})$.
    end if
end for
return $\hat{q} \leftarrow q^{(T+1)}$.
```

with overwhelming probability, all clean samples are retained. The constant factors hidden in the hyper-parameters $b_k, r_k, \tau_k, \delta_k, k_{\max}$ can be found in Appendix C.

2.2 REWEIGHT: Localized Soft Outlier Removal

We elaborate on the key idea to solve (1) in nearly linear time, assuming the existence of a feasible solution. In particular, we assume that $q^* = (q_1^*, \dots, q_n^*)$ is feasible, where $q_i^* = \frac{1}{n}$ if (x_i, y_i) is a clean sample and $q_i^* = 0$ otherwise. Note that such feasibility is guaranteed as far as the sample size $n \geq d \cdot \mathrm{polylog}(d, \frac{1}{b}, \frac{1}{\delta})$ (see Theorem 7 of Shen (2021b)). For a weight vector $q \geq 0$, we will frequently denote the reweighted empirical covariance matrix[2] by

$$M(q) := \sum_{i=1}^n q_i x_i x_i^\top. \tag{3}$$

Since Eq. (1) is a linear program, a natural solver is the ellipsoid method. It, however, turns out that the computational bottleneck of the ellipsoid method roots in its high iteration complexity, i.e. $O(d^2)$ (Bubeck, 2015), and the per-iteration cost. In particular, in each iteration, the method identifies one direction corresponding to the maximum eigenvalue of the empirical covariance matrix to construct the separation oracle, which runs in time $O(nd^2 + d)$. An interesting observation made in Dong et al. (2019); Hopkins et al. (2020) is that a relevant problem, robust mean estimation, can be solved via online regret minimization (Cesa-Bianchi et al., 2004; Hazan, 2019). Though that is an unsupervised learning problem while we consider a supervised setting, it turns out that they share merit in dealing with corrupted samples but with two key differences: first, the empirical covariance matrix of those works involves the empirical mean; second, our samples are instance-label pairs with instances being drawn from an isotropic log-concave distribution conditioned on a band while their samples are drawn from Gaussian or sub-gaussian. As will be clear, these lead to different design on loss functions and analysis.

We now turn to the design of the algorithm. We follow Dong et al. (2019) and appeal to the matrix multiplicative weights update (MMWU) method (Arora et al., 2012) to solve (1) by invoking REFINE (Algorithm 3) that takes as input an initial weight vector $0 < q < \frac{1}{n}$ and uses MMWU to iteratively improve the weight vector in the sense of reducing the spectral norm of $M(q)$, and outputs a new vector q' with $\|M(q')\| \leq \frac{3}{4}\|M(q)\|$. To this end, we will need a result from Allen-Zhu et al. (2015) which states that

[2] We slightly abuse the terminology "covariance matrix" in the paper by referring to the one without subtracting the mean.

for any sequence $\{q^{(1)}, \dots, q^{(T+1)}\}$,

$$\Big\|\sum_{t=1}^{T} M(q^{(t+1)})\Big\| \le \sum_{t=1}^{T} \langle M(q^{(t+1)}), U^{(t)} \rangle + \lambda \sum_{t=1}^{T} \langle M(q^{(t+1)}), U^{(t)} \rangle \|M(q^{(t+1)})\| + \frac{\log d}{\lambda} \tag{4}$$

holds for any $\lambda \le \min_{t \in [T]} \|M(q^{(t+1)})\|^{-1}$ where

$$U^{(t)} := \frac{\exp(\lambda \sum_{s=1}^{t} M(q^{(s)}))}{\operatorname{tr} \exp(\lambda \sum_{s=1}^{t} M(q^{(s)}))}. \tag{5}$$

The underlying intuition for the choice of $U^{(t)}$ is that it is the global optimum of the following entropy-regularized semidefinite program:

$$\max_{U \in \mathbb{R}^{d \times d}} \lambda \Big\langle \sum_{s=1}^{t} M(q^{(s)}), U \Big\rangle + \langle U, -\log U \rangle, \quad \text{s. t. } U \succeq 0,\ \operatorname{tr}(U) = 1. \tag{6}$$

This is a follow-the-regularized-leader type objective function with entropy regularization (Shalev-Shwartz, 2012) in the matrix case. When $\lambda \to \infty$, we can see that the optimal U is given by $U = vv^\top$ with v being the top eigenvector of the sum of $M(q^{(s)})$, namely, it reduces to the approach of finding one direction in one step; when $\lambda \to 0$, $U = \frac{1}{d} I_{d \times d}$ which weighs all directions equally. Therefore, a positive finite λ induces a solution U that interpolates between the two extremes, thus will find several principal directions. In terms of iteration complexity, it is possible to show that when the aggregated variance $\langle M(q^{(t)}), U^{(t)} \rangle$ is large, we can apply the 1D-FILTER of Dong et al. (2019) to reduce it. In this way, the right-hand side of (4) is at most $(\frac{1}{2}T + \log d) \cdot \|M(q^{(1)})\|$. This shows that after $T = 4 \log d$ iterations, REFINE can output a weight vector $q^{(T+1)}$ such that $\|M(q^{(T+1)})\| \le \frac{3}{4} \|M(q^{(1)})\|$.

However, calculating the matrix $U^{(t)}$ is computationally expensive. To avoid this, we observe that

$$\langle M(q^{(t+1)}), U^{(t)} \rangle = \sum_{i=1}^{n} q_i^{(t+1)} x_i^\top U^{(t)} x_i =: \sum_{i=1}^{n} q_i^{(t+1)} \beta_i^{(t)}. \tag{7}$$

It turns out that one can compute efficiently an approximation to $\beta_i^{(t)}$ for all $i \in [n]$:

Lemma 7 (Lemma 5.1 of Dong et al. (2019)). *Consider any fixed iteration t of* REFINE. *There exists an algorithm* $\mathrm{MMWUScore}(S, q^{(1)}, \dots, q^{(t)}, \delta)$ *that runs in time $\tilde{O}(tnd \log \frac{1}{\delta})$ and with probability $1 - \delta$, outputs a non-negative vector $\tilde{\beta}^{(t)}$ such that $\tilde{\beta}_i^{(t)} / \beta_i^{(t)} \in [\frac{9}{10}, \frac{11}{10}]$.*

Now we are in the position to state the main idea of REWEIGHT: we start with the uniform distribution $p^{(1)} = (\frac{1}{n}, \dots, \frac{1}{n})$ on S, and repeatedly invoke the subroutine REFINE to produce a more accurate empirical distribution in each iteration j such that the spectral norm of $M(p^{(j+1)})$ is at most $\frac{3}{4}$ of before. Therefore, after $O\big(\log \frac{\|M(p^{(1)})\|}{\lambda^*}\big)$ iterations, we can find a distribution $\hat{p}$ under which the spectral norm is less than a prescribed bound λ^* (up to a constant factor). Since we are promised that the ℓ_2-norm of all x_i in S is bounded by γ, it is not hard to show that $\|M(p^{(1)})\| \le \gamma^2$. This gives the setting of the maximum iteration number J in Algorithm 2.

A minor implementation detail of REWEIGHT is that for the sake of computational efficiency, we will use a randomized algorithm $\mathrm{APPROXEV}(S, q, \alpha, \delta)$ that can efficiently approximate the maximum eigenvalue of a given symmetric matrix $M = \sum_{i=1}^{n} q_i x_i^\top x_i$. It is known that the classic power method suffices and runs in linear-time provided that the target approximation error is a constant.

Lemma 8 (Kuczynski and Wozniakowski (1992)). *There exists an algorithm* APPROXEV *that takes as input $S \subset \mathbb{R}^d$, $q \ge 0$, $\alpha \in (0, 1)$, $\delta \in (0, 1)$, and runs in time $O(\frac{nd}{\alpha} \log \frac{1}{\delta})$ and outputs a scalar $\lambda > 0$ such that $\lambda / \|M(q)\| \in [1 - \alpha, 1 + \alpha]$ where $M(q) := \sum_{i=1}^{n} q_i x_i^\top x_i$.*

Note also that before invoking REFINE, we will check if the obtained approximate eigenvalue $\lambda^{(j)}$ is less than a constant factor of λ^*. If this is the case, the algorithm will return the current weight vector $p^{(j)}$. Such simple check not only potentially saves computational cost, but also appears useful to derive the performance guarantee of REFINE.

Lastly, for technical reason, in REFINE, we impose the output weight vector $\hat{q}$ in Q_S, where

$$Q_S := \Big\{ q \in \mathbb{R}^n : 0 \le q \le \frac{1}{n}, \sum_{i \in S_{\mathrm{C}}} \Big(\frac{1}{n} - q_i\Big) \le \sum_{i \in S_{\mathrm{D}}} \Big(\frac{1}{n} - q_i\Big) \Big\}, \tag{8}$$

and S_{C} and S_{D} denote the set of clean and corrupted samples in S respectively. This constraint set Q_S is different from the one in Eq. (1); yet it is not hard to see that they share the same merit of enforcing the algorithm to assign large weights to clean samples. In fact, the second constraint above ensures that the weights of corrupted samples are decreased faster than those of clean samples, an important property to characterize the dynamic of the REFINE algorithm.

2.3 OPTIMIZE: Hinge Loss Minimization

In order to obtain nearly linear-time algorithm to optimize (2) up to a constant optimization error $\kappa > 0$, we appeal to the stochastic gradient descent (SGD) algorithm, whose iteration complexity is higher than gradient descent but the per-iteration cost is independent of the sample size (Nesterov, 2004; Bubeck, 2015). Additionally, we also hope to

Algorithm 4 OPTIMIZE

```
Require: A sample set $S = \{(x_i, y_i)\}_{i=1}^n$, probability distribution $p = (p_1, \dots, p_n)$ on $S$, constraint set $W = \{w : \|w - u\| \leq r\}$, bandwidth $b$ such that $\max_{i \in [n]} |u \cdot x_i| \leq b$, scalar $\gamma$ such that $\max_{i \in [n]} \|x_i\| \leq \gamma$, failure probability $\delta'$.
Ensure: A hypothesis $\hat{v}$ such that $\ell_\tau(\hat{v}; S, p) \leq \min_{w \in W} \ell_\tau(w; S, p) + \kappa$.
$v_0 \leftarrow 0$ and $T \leftarrow \frac{1}{\kappa^2} \cdot \frac{72(b+r)^2\gamma^2}{\tau^2} \cdot \log \frac{4}{\delta'}$.
for $t = 1, 2, \dots, T$ do
    Draw $(x_{i_t}, y_{i_t})$ from $S$ according to $p$.
    $v_t \leftarrow v_{t-1} + \frac{\rho_t}{\tau} y_{i_t} x_{i_t} \cdot \mathbf{1}\{y_{i_t} v_{t-1} \cdot x_{i_t} < \tau\}$ where $\rho_t = \frac{2r\tau}{\gamma\sqrt{t}}$.
    if $v_t \notin W$, $v_t \leftarrow u + \frac{v_t - u}{\|v_t - u\|} \cdot r$.
end for
return $\hat{v} \leftarrow \frac{1}{T} \sum_{t=1}^T v_t$.
```

obtain a high probability convergence argument for SGD rather than convergence in expectation. While there have been such results for strongly convex objective functions (Shalev-Shwartz et al., 2011), to the best of our knowledge, no general analysis has been established for applying SGD on convex and Lipschitz functions (which is our case).

Our workaround is a two-step analysis: first, we think of SGD as applying online gradient descent (Zinkevich, 2003) to a sequence of hinge loss functions $f_{i_t}(w) = \max\{0, 1 - \frac{1}{\tau} y_{i_t} w \cdot x_{i_t}\}$ where the index i_t is chosen according to the distribution p on S rather than being adversarially; second, we apply online-to-batch conversion (Cesa-Bianchi et al., 2004) to obtain a high probability guarantee on the optimization performance.

Of central importance of online gradient descent and other online optimization algorithms is the notion of regret, which measures the cumulative loss incurred during online updates compared to the smallest possible loss at hindsight. The regret depends on the diameter of the constraint set W and the magnitude of the gradient of the loss functions. Thanks to Step 7 of Algorithm 1, the ℓ_2-norm of all the instances is well-controlled. This gives the learning rate that we use in OPTIMIZE. We show that this also implies a regret bound of $O(\frac{r\gamma}{\tau}\sqrt{T})$ using standard results; see, e.g. Theorem 3.1 of Hazan (2019). We further upper bound the loss function as $O(\frac{b+r}{\tau}\gamma)$. Putting these pieces and the guarantee of online-to-batch conversion together, we obtain the following result for OPTIMIZE:

Theorem 9 (OPTIMIZE). *Consider Algorithm 4. With probability $1 - \delta'$, the following holds: the algorithm runs in time $O(Td)$ and its output $\hat{v}$ is such that $\ell_\tau(\hat{v}; S, p) \leq \min_{w \in W} \ell_\tau(w; S, p) + \kappa$. In particular, under the setting $b = \Theta(r) = \Theta(\tau)$ and $\kappa = \Theta(1)$, the computational complexity is $O(d^2 \cdot \log^3(\frac{n}{b\delta'}))$.*

3 PERFORMANCE GUARANTEES

We will fix a phase k throughout the section and drop the subscript k. We will use u in place of w_{k-1}. Our goal is to find a feasible solution to (1). We note, however, that we will consider a sufficient condition for the last constraint thereof to ease the algorithmic design.

Lemma 10. *Let $V = \{v \in \mathbb{R}^d : \|v\| \leq 1\}$. Suppose that a weight vector $q \in \mathbb{R}^n$ is such that $0 \leq q \leq \frac{1}{n}$, $\sum_{i=1}^n q_i \geq 1 - \xi$, and $\sup_{v \in V} \sum_{i=1}^n q_i (v \cdot x_i)^2 = O(1)$. Then q is feasible to (1).*

Our algorithm was thus designed to solve the following program:

$$\begin{aligned} &\min_{q = (q_1, \dots, q_n)} \ \langle 0, q \rangle \\ &\text{s.t. } 0 \leq q \leq \frac{1}{n}, \ \sum_{i=1}^n q_i \geq 1 - \xi, \\ &\sup_{v : \|v\| \leq 1} \sum_{i=1}^n q_i (v \cdot x_i)^2 \leq O(1). \end{aligned} \tag{9}$$

The following lemma shows that the ideal solution, that $q_i^* = \frac{1}{n}$ if (x_i, y_i) is clean sample (drawn from $D_{u,b}$) and $q_i^* = 0$ otherwise, is feasible.

Lemma 11. *Consider the set of samples $S = \{(x_i, y_i)\}_{i=1}^n$ obtained at Step 7 of Algorithm 1 at any phase. Let $q_i^* = \frac{1}{n}$ if (x_i, y_i) is clean sample and $q_i^* = 0$ otherwise. Then with probability $1 - \delta$, $\|M(q^*)\| \leq \lambda^*$ for some absolute constant $\lambda^* > 0$ if $n \geq d \cdot \mathrm{polylog}(d, \frac{1}{b}, \frac{1}{\delta})$.*

The following is the primary result for soft outlier removal under the feasibility of q^*.

Theorem 12 (REWEIGHT). *Consider the* REWEIGHT *algorithm (i.e. Algorithm 2). Let $\xi \in [0, \frac{1}{2})$ be the fraction of corrupted samples in S. With probability $1 - \delta'$, the output $\hat{p}$ of the algorithm is such that $0 \leq \hat{p} \leq \frac{1}{n}$, $\|\hat{p}\|_1 \geq 1 - 2\xi$, and $\|M(\hat{p})\| \leq 1350\lambda^*$. The running time is $\tilde{O}\big(nd \cdot \log^2(\frac{\gamma}{\lambda^* \delta'})\big)$.*

Observe that $\hat{p}$ slightly violates the constraint in (9) as $\|\hat{p}\|_1 \geq 1 - 2\xi$. As will be clear in our analysis, this does not hurt our main results.

Of the core of the analysis of Theorem 12 is a performance guarantee of REFINE. Recall $S = S_{\mathrm{C}} \cup S_{\mathrm{D}}$ where S_{C} consists of samples drawn from $D_{u,b}$ and S_{D} is the set of corrupted samples, and recall the constraint set Q_S defined in (8). It is easy to show the following:

Lemma 13. *For any $q \in Q_S$, $\|q\|_1 \geq 1 - \frac{2|S_{\mathrm{D}}|}{n}$. In particular, when $|S_{\mathrm{D}}| \leq \xi n$, $\|q\|_1 \geq 1 - 2\xi$.*

Therefore, during the updates within REFINE, we will maintain the iterates such that they lie in Q_S.

Theorem 14 (REFINE). *Consider the* REFINE *algorithm (i.e. Algorithm 3) where the input $q \in Q_S$. With probability $1 - \delta''$, the output $\hat{q}$ is in Q_S, and satisfies $\|M(\hat{q})\| \leq \frac{3}{4}\|M(q)\|$ or $\|M(\hat{q})\| \leq 1350\lambda^*$. The running time is $\tilde{O}\big(nd \cdot \log(\frac{\gamma}{\lambda^* \delta''})\big)$.*

We are now in the position to prove Theorem 3.

Proof of Theorem 3. Consider Algorithm 1. In each phase k, we perform the rejection sampling (Step 6) to gather n_k samples in X_k. Lemma 20 shows that this can be done by making a number of $m_k = \Omega(\frac{1}{b_k}(n_k + \log\frac{1}{\delta}))$ calls to $\mathrm{EX}(D, w^*)$, leading to a computational complexity bound of $O(m_k)$. The pruning step (Step 7) checks the ℓ_2-norm of all instances in S, leading to a computational complexity bound of $O(n_k d)$. Next, by Lemma 22, $\gamma_k = \sqrt{d}\log\big(\frac{4en_k}{c_8 b_k \delta_k}\big) \leq \tilde{O}(\sqrt{d}\log\frac{1}{\epsilon\delta})$. In addition, Lemma 11 tells that λ^* is an absolute constant. Thus, the running time of REWEIGHT is $\tilde{O}\big(n_k d \cdot \log^2(\frac{1}{\epsilon\delta})\big)$ in view of Theorem 12 and that of OPTIMIZE is $\tilde{O}\big(d^2 \cdot \log^3(\frac{1}{\epsilon\delta})\big)$ in view of Theorem 9. As $m_k \geq n_k \geq d$, it is easy to see that the running time for phase k is $\tilde{O}\big(m_k d \cdot \log^3(\frac{1}{\epsilon\delta})\big)$. Summing over $k_{\max} = \Theta(\log\frac{1}{\epsilon})$ phases, we conclude that the overall running time of Algorithm 1 is $\tilde{O}\big(m_{\mathcal{A}} d \cdot \log^4(\frac{1}{\epsilon\delta})\big)$ where $m_{\mathcal{A}} := \sum_{k=1}^{k_{\max}} m_k = \frac{d}{\epsilon} \cdot \mathrm{polylog}(d, \frac{1}{\epsilon}, \frac{1}{\delta})$.

The sample complexity bound and PAC guarantee of the output $\hat{w}$ of Algorithm 1 directly follows from that of Shen (2021b), since in order to fit their analysis, we only need to find global optimum of (9) and (2).This completes the proof. □

4 CONCLUSION AND OPEN QUESTIONS

We investigated the problem of learning homogeneous halfspaces in the presence of the malicious noise. This is an important problem that has been broadly studied in learning theory, typically with the emphasis on understanding noise tolerance and sample complexity of polynomial-time algorithms. We have presented the first nearly linear-time algorithm that achieves the best along all dimensions: computational complexity, sample complexity, label complexity, and noise tolerance. To this end, we leveraged the powerful and general matrix multiplicative weights update method into a margin-based active learning framework. We also presented a two-step high probability analysis of stochastic gradient descent for non-smooth and non-strongly convex functions.

We believe that our techniques can find more applications, such as fast learning of generalized linear models with the malicious noise. One open question that we have in mind is whether it is possible to extend our algorithm to learning of sparse halfspaces with the same noise. It turns out that with the sparsity constraint, it becomes intractable to evaluate the last constraint of the underlying soft outlier removal program (1), due to the computational hardness of sparse principal component analysis. As far as we know, recent works of Diakonikolas et al. (2019b); Shen and Zhang (2021) only lead to super-linear time algorithms, and it is unclear how we can adapt the MMWU framework with the sparsity constraint; in particular, how to efficiently compute the global optimum of (6) when there is an additional sparsity constraint on U.

Acknowledgements

We thank the anonymous reviewers and meta-reviewer for valuable comments. This work is supported by NSF-IIS-1948133 and by the startup funding from Stevens Institute of Technology.

References

Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 237–245, 2015.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Proceedings of the 28th Annual Conference on Learning Theory*, pages 167–190, 2015.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 152–192, 2016.

Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):50:1–50:27, 2017.

Sivaraman Balakrishnan, Simon S. Du, Jerry Li, and Aarti Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 169–212, 2017.

Maria-Florina Balcan and Philip M. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the 26th Annual Conference on Learning Theory*, pages 288–316, 2013.

Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, pages 35–50, 2007.

Avrim Blum, Alan M. Frieze, Ravi Kannan, and Santosh S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 330–338, 1996.

Nader H. Bshouty, Nadav Eiron, and Eyal Kushilevitz. PAC learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

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

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9): 2050–2057, 2004.

Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Classification under misspecification: Halfspaces, generalized linear models, and evolvability. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, 2020.

Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2755–2771, 2019.

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

Ilias Diakonikolas and Daniel M. Kane. Recent advances in algorithmic high-dimensional robust statistics. *CoRR*, abs/1911.05911, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 655–664, 2016.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 999–1008, 2017a.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science*, pages 73–84, 2017b.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, pages 1061–1073, 2018.

Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent PAC learning of halfspaces with Massart noise. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 4751–4762, 2019a.

Ilias Diakonikolas, Daniel Kane, Sushrut Karmalkar, Eric Price, and Alistair Stewart. Outlier-robust high-dimensional sparse estimation via iterative filtering. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 10688–10699, 2019b.

Ilias Diakonikolas, Daniel Kane, Sushrut Karmalkar, Eric Price, and Alistair Stewart. Outlier-robust high-dimensional sparse estimation via iterative filtering. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 10688–10699, 2019c.

Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. A polynomial time algorithm for learning halfspaces with Tsybakov noise. *CoRR*, abs/2010.01705, 2020a.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with Massart noise under structured distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory*, pages 1486–1513, 2020b.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with tsybakov noise. *CoRR*, abs/2006.06467, 2020c.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustness meets algorithms. *Communications of the ACM*, 64(5):107–115, 2021a.

Ilias Diakonikolas, Daniel M. Kane, Thanasis Pittas, and Nikos Zarifis. The optimality of polynomial regression for agnostic learning under gaussian marginals in the SQ model. In *Proceedings of the 34th Conference on Learning Theory*, pages 1552–1584, 2021b.

Yihe Dong, Samuel B. Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. volume abs/1906.11366, 2019.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–574, 2006.

Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 543–552, 2006.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019.

Samuel B. Hopkins, Jerry Li, and Fred Zhang. Robust and heavy-tailed mean estimation made simple, via regret minimization. *CoRR*, abs/2007.15839, 2020.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 11–20, 2005.

Michael J. Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 267–280, 1988.

Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. In *Proceedings of the 5th Annual Conference on Computational Learning Theory*, pages 341–352, 1992.

Adam R. Klivans, Philip M. Long, and Rocco A. Servedio. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10:2715–2740, 2009.

Jacek Kuczynski and Henryk Wozniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

Sanjeev R. Kulkarni, Sanjoy K. Mitter, and John N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–35, 1993.

Kevin A. Lai, Anup B. Rao, and Santosh S. Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 665–674, 2016.

László Lovász and Santosh S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures and Algorithms*, 30(3):307–358, 2007.

Wolfgang Maass and György Turán. How fast can a threshold gate learn? In *Proceedings of a workshop on computational learning theory and natural learning systems (vol. 1): constraints and prospects*, pages 381–414, 1994.

Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, pages 2326–2366, 2006.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer US, 2004.

Frank Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

Rocco A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1): 3–30, 2011.

Jie Shen. On the power of localized Perceptron for label-optimal learning of halfspaces with adversarial noise. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9503–9514, 2021a.

Jie Shen. Sample-optimal PAC learning of halfspaces with malicious noise. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9515–9524, 2021b.

Jie Shen and Chicheng Zhang. Attribute-efficient learning of halfspaces with malicious noise: Near-optimal label complexity and noise tolerance. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, pages 1072–1113, 2021.

Robert H. Sloan. Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, pages 91–96, 1988.

Alexander B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166, 2004.

John W. Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

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

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

Santosh S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *Journal of the ACM*, 57(6):32:1–32:14, 2010.

Songbai Yan and Chicheng Zhang. Revisiting Perceptron: Efficient and label-optimal learning of halfspaces. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 1056–1066, 2017.

Shiwei Zeng and Jie Shen. List-decodable sparse mean estimation. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, 2022.

Chicheng Zhang and Yinan Li. Improved algorithms for efficient active learning halfspaces with Massart and Tsybakov noise. In *Proceedings of the 34th Annual Conference on Learning Theory*, pages 4526–4527, 2021.

Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pages 7184–7197, 2020.

Lijun Zhang, Jinfeng Yi, and Rong Jin. Efficient algorithms for robust one-bit compressive sensing. In *Proceedings of the 31st International Conference on Machine Learning*, pages 820–828, 2014.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

A OMITTED PROOFS

A.1 Proof of Proposition 2

Proof. The proof is rather straightforward. Under the assumption that D is standard Gaussian, it is a folklore that for any clean sample (x_i, y_i) returned by $\mathrm{EX}(D, w^*)$,

$$\mathbb{E}_{x \sim D}[y_i x_i] = \mathbb{E}_{x \sim D}[\mathrm{sign}(w^* \cdot x_i) \cdot x_i] = \sqrt{\frac{2}{\pi}} w^*; \tag{10}$$

see for example, Lemma 4 of Zhang et al. (2014). Therefore, $z_i := \sqrt{\pi/2} y_i x_i$ are independent random draws from some sub-gaussian distribution D' with unknown mean w^* and identity covariance matrix for all clean samples (x_i, y_i), and the rest of z_i are arbitrary. This is exactly the problem setup of robust mean estimation considered in Lemma 23. Thus, we directly apply Lemma 23 to get that there is an algorithm that runs in time $\tilde{O}(nd)$ and with probability $1 - \delta$, outputs $\hat{w} \in \mathcal{H}$ such that

$$\|\hat{w} - w^*\| \leq O\Big(\eta \sqrt{\log(1/\eta)} + \sqrt{\frac{d + \log(1/\delta)}{n}}\Big).$$

It is easy to see that when $n \geq \Omega\big(\frac{d + \log(1/\delta)}{\epsilon^2}\big)$ and $\eta \leq O\big(\frac{\epsilon}{\log(1/\epsilon)}\big)$, we have $\|\hat{w} - w^*\| \leq K \cdot \epsilon$ for sufficiently small constant $K > 0$.

Now by Balcan and Long (2013), when the distribution D is isotropic log-concave (which applies to the Gaussian), we have

$$\mathrm{Pr}_{x \sim D}\left(\mathrm{sign}(\hat{w} \cdot x) \neq \mathrm{sign}(w^* \cdot x)\right) \leq O(\theta(\hat{w}, w^*)) \leq O(\|\hat{w} - w^*\|) \leq \epsilon, \tag{11}$$

where the last step follows since we can choose K sufficiently small. The proof is complete. □

A.2 Proof of Lemma 10

Proof. It is not hard to show that

$$\sup_{w \in W} \sum_{i=1}^n p_i (w \cdot x_i)^2 \leq 2 \sum_{i=1}^n q_i (u \cdot x_i)^2 + 2 \sup_{w \in W} \sum_{i=1}^n q_i ((w - u) \cdot x_i)^2.$$

By localized sampling, we have $(u \cdot x_i)^2 \leq b^2$ for all x_i; thus the first term on the right-hand side is at most $2b^2$. Since $W = \{w : \|w - u\| \leq r\}$, the second term on the right-hand side equals $2r^2 \sup_{v \in V} \sum_{i=1}^n q_i (v \cdot x_i)^2$ in view of simple variable change. □

A.3 Proof of Lemma 11

Proof. Denote S'_{C} the set of clean samples in S' and S_{C} that of S.

By Lemma 20, if we call $\mathrm{EX}(D, w^*)$ for $m \geq \Omega\big(\frac{1}{b}(n + \log\frac{3}{\delta})\big)$ times, with probability $1 - \frac{\delta}{4}$, we can form S' with size n'. Next, by Lemma 21, the number of clean samples in S' is at least $\frac{3}{4}\left|S'\right| = \frac{3}{4}n'$. Now by Lemma 22, we know that with probability $1 - \frac{\delta}{4}$, all clean samples are retained during Step 7 of Algorithm 1. Thus, $|S_{\mathrm{C}}| \geq \frac{3}{4}n' \geq \frac{3}{4}n$. Therefore, as soon as we set $n \geq d \cdot \mathrm{polylog}(d, \frac{1}{b}, \frac{1}{\delta})$, we can apply Lemma 19 and complete the proof. □

A.4 Analysis of REWEIGHT

Suppose $\lambda = \textsc{ApproxEV}(S, q, 1/10, \delta)$ and $\lambda' = \textsc{ApproxEV}(S, q', 1/10, \delta)$. We will frequently use the following simple facts from algebraic calculation:

$$\lambda' \leq \zeta\lambda \Longrightarrow \|M(q')\| \leq \frac{11}{9}\zeta\|M(q)\|, \quad \|M(q')\| \leq \zeta\|M(q)\| \Longrightarrow \lambda' \leq \frac{11}{9}\zeta\lambda. \tag{12}$$

Let $S = S_{\mathrm{C}} \cup S_{\mathrm{D}}$ where S_{C} consists of clean samples and S_{D} is the set of corrupted samples. Recall the definition:

$$Q_S = \left\{ q \in \mathbb{R}^n : 0 \leq q \leq \frac{1}{n}, \sum_{i \in S_{\mathrm{C}}} \Big(\frac{1}{n} - q_i\Big) \leq \sum_{i \in S_{\mathrm{D}}} \Big(\frac{1}{n} - q_i\Big) \right\}.$$

Lemma 15 (Restatement of Lemma 13)**.** *For any $q \in Q_S$, $\|q\|_1 \geq 1 - \frac{2\,|S_{\mathrm{D}}|}{n}$. In particular, when $|S_{\mathrm{D}}| \leq \xi n$, $\|q\|_1 \geq 1-2\xi$.*

Proof. By the last constraint in (8) and the fact $q \geq 0$, we have

$$\sum_{i=1}^{n}\Big(\frac{1}{n}-q_i\Big) \leq 2\sum_{i\in S_{\mathrm{D}}}\Big(\frac{1}{n}-q_i\Big) \leq \frac{2|S_{\mathrm{D}}|}{n}.$$

Rearranging gives the desired bound on $\|q\|_1$. □

Let q^* be such that $q^*_i = \frac{1}{n}$ if $i \in S_{\mathrm{C}}$ and $q^*_i = 0$ otherwise. Clearly, $q^* \in Q_S$ and

$$\|M(q^*)\| = \frac{1}{n}\|\sum_{i\in S_{\mathrm{C}}} x_i^\top x_i\| \leq \frac{|S_{\mathrm{C}}|}{n}\cdot\lambda^* \leq \lambda^*. \tag{13}$$

A.4.1 Proof of Theorem 12

Proof. By the construction of $p^{(1)}$, we have $p^{(1)} \in Q_S$. Thus the guarantee from Theorem 14 ensures $p^{(j+1)} \in Q_S$ for all $1 \leq j \leq J$.

In addition, for any $1 \leq j \leq J$, Theorem 14 tells that either $\|M(p^{(j+1)})\| \leq \frac{3}{4}\|M(p^{(j)})\|$ or $\|M(p^{(j+1)})\| \leq 1350\lambda^*$. In the latter case, we will have $\lambda^{(j+1)} \leq 1500\lambda^*$, for which REWEIGHT will return. Now consider such case does not occur for all j. Then by telescoping the inequality $\|M(p^{(j+1)})\| \leq \frac{3}{4}\|M(p^{(j)})\|$, we must have

$$\|M(p^{(J+1)})\| \leq \Big(\frac{3}{4}\Big)^J\|M(p^{(1)})\| = \Big(\frac{3}{4}\Big)^J \max_{v:\|v\|=1}\frac{1}{n}\sum_{i=1}^{n}(v\cdot x_i)^2 \leq \Big(\frac{3}{4}\Big)^J\cdot\gamma^2.$$

With the choice $J = \log_{4/3}(\frac{\gamma^2}{\lambda^*})$, we obtain $\|M(p^{(J+1)})\| \leq \lambda^*$.

In either case, we know that the returned (unnormalized) p satisfies $\|M(p)\| \leq 1350\lambda^*$.

Since the running time of REFINE is $\tilde{O}(nd\log\frac{\gamma J}{\lambda^*\delta'})$, the total running time of REWEIGHT is J times more. Given the setting of J, it is not hard to see that the computational complexity is $\tilde{O}(nd\log^2\frac{\gamma}{\lambda^*\delta'})$.

Finally, each time REFINE may fail with probability $\frac{\delta'}{J}$. Thus, by union bound over all J iterations, the REWEIGHT algorithm may fail with probability at most δ'. □

A.5 Analysis of REFINE

Lemma 16 (Progress within REFINE)**.** *Suppose that $q \in Q_S$ and $\|M(q)\| \geq 1000\|M(q^*)\|$ where $q^*_i = \frac{1}{n}$ if $i \in S_{\mathrm{C}}$ and $q^*_i = 0$ otherwise. Let $U \in \Delta_{d\times d}$, $\beta_i = x_i^\top U x_i$ and $\tilde{\beta}_i$ be such that $\tilde{\beta}_i/\beta_i \in [\frac{9}{10},\frac{11}{10}]$ for all $i \in [n]$. Let λ be such that $\lambda/\|M(q)\| \in [\frac{9}{10},\frac{11}{10}]$, and $\langle q,\tilde{\beta}\rangle \geq \frac{1}{5}\lambda$. Then, for $q' = $ 1D-FILTER$(q,\tilde{\beta},\frac{1}{90})$, we have $0 \leq q' \leq q$, $q' \in Q_S$, and $\langle M(q'),U\rangle \leq \frac{1}{16}\langle M(q),U\rangle$. Furthermore, the running time is $O\big(n\log\frac{\|\beta\|_\infty}{\langle q,\beta\rangle}\big)$.*

Proof. The proof of the lemma follows from Claim 3.7 of Dong et al. (2019), though our proof is simpler since we do not need to estimate the mean of the distribution.

We first verify that $q \in Q_S$ satisfies the assumption of Lemma 24, i.e. $\langle q_{S_{\mathrm{C}}},\tilde{\beta}_{S_{\mathrm{C}}}\rangle \leq \alpha\langle q,\tilde{\beta}\rangle$ for some factor $\alpha > 0$. In fact, since $q \in Q_S$, we have $q_i \leq \frac{1}{n}$ for all $i \in [n]$. It follows that

$$\langle q_{S_{\mathrm{C}}},\beta_{S_{\mathrm{C}}}\rangle \leq \langle q^*,\beta\rangle \leq \|M(q^*)\| \leq \frac{1}{1000}\|M(q)\| \leq \frac{\lambda}{900} \leq \frac{1}{180}\langle q,\tilde{\beta}\rangle, \tag{14}$$

where the second step follows from the definition of spectral norm, the third step follows from our assumption on $\|M(q)\|$, the last two steps follow from our assumptions on λ. On the other hand, since we assumed for all $i \in [n]$ that $\beta_i \geq \frac{10}{11}\tilde{\beta}_i$ and q_i is positive, we have $\langle q_{S_{\mathrm{C}}},\beta_{S_{\mathrm{C}}}\rangle \geq \frac{10}{11}\langle q_{S_{\mathrm{C}}},\tilde{\beta}_{S_{\mathrm{C}}}\rangle$. This together with the above inequality implies

$$\langle q_{S_{\mathrm{C}}},\tilde{\beta}_{S_{\mathrm{C}}}\rangle \leq \frac{11}{1800}\langle q,\tilde{\beta}\rangle \leq \frac{1}{90}\langle q,\tilde{\beta}\rangle.$$

Therefore, we can invoke Lemma 24 by passing the arguments $(q, \tilde{\beta}, \frac{1}{45})$ to 1D-FILTER and obtain a weight vector q' such that

$$\langle q', \tilde{\beta} \rangle \leq \frac{1}{45} \langle q, \tilde{\beta} \rangle.$$

By the condition $\tilde{\beta}_i / \beta_i \in [\frac{9}{10}, \frac{11}{10}]$, we convert back to the guarantee in terms of β as follows:

$$\langle q', \beta \rangle \leq \frac{1}{45} \cdot \frac{11}{9} \langle q, \beta \rangle \leq \frac{1}{16} \langle q, \beta \rangle.$$

Observe that this is equivalent to

$$\langle M(q'), U \rangle \leq \frac{1}{16} \langle M(q), U \rangle. \tag{15}$$

In addition, the first part of Lemma 24 gives $0 \leq q' \leq q$ and $\sum_{i \in S_{\mathrm{C}}} (q_i - q'_i) \leq \sum_{i \in S_{\mathrm{D}}} (q_i - q'_i)$. It hence follows that $0 \leq q' \leq \frac{1}{n}$ since $q \leq \frac{1}{n}$. Also,

$$\sum_{i \in S_{\mathrm{C}}} \left(\frac{1}{n} - q'_i \right) = \sum_{i \in S_{\mathrm{C}}} \left(\frac{1}{n} - q_i \right) + \sum_{i \in S_{\mathrm{C}}} (q_i - q'_i) \leq \sum_{i \in S_{\mathrm{D}}} \left(\frac{1}{n} - q_i \right) + \sum_{i \in S_{\mathrm{D}}} (q_i - q'_i) = \sum_{i \in S_{\mathrm{D}}} (q_i - q'_i),$$

namely, we proved $q' \in Q_S$.

Lastly, the running time of 1D-FILTER is given by $O\Big(n \log \frac{45 \|\tilde{\beta}\|_\infty}{\langle q, \tilde{\beta} \rangle}\Big)$. Since β differs from $\tilde{\beta}$ by a constant factor, this amounts to $O\Big(n \log \frac{\|\beta\|_\infty}{\langle q, \beta \rangle}\Big)$. □

Next, we characterize the performance of the $\mathrm{MMWUScore}$ subroutine in REFINE. For an iteration t, recall

$$U^{(t)} := \frac{\exp(\lambda \sum_{s=1}^{t} M(q^{(s)}))}{\operatorname{tr} \exp(\lambda \sum_{s=1}^{t} M(q^{(s)}))}, \quad \beta_i^{(t)} := x_i^\top U^{(t)} x_i \text{ for all } i \in [n], \text{ where } \lambda = \frac{9}{10 \lambda^{(1)}}.$$

Recall also that Lemma 7 shows that there exists an algorithm that computes $\tilde{\beta}^{(t)}$ in nearly linear time with $\tilde{\beta}_i^{(t)} / \beta_i^{(t)} \in [\frac{9}{10}, \frac{11}{10}]$ for all $t \in [T]$ and $i \in [n]$.

Now we prove the main result for REFINE.

A.5.1 Proof of Theorem 14

Proof. First, if REFINE returns at Step 4 in some iteration t, we have $\lambda^{(t)} \leq \frac{1}{2} \lambda^{(1)}$ or $\lambda^{(t)} \leq 1200 \lambda^*$. Recall $\lambda^{(t)} / \|M(q^{(t)})\| \in [\frac{9}{10}, \frac{11}{10}]$. If $\lambda^{(t)} \leq \frac{1}{2} \lambda^{(1)}$,

$$\|M(q^{(t)})\| \leq \frac{11}{9} \cdot \frac{1}{2} \|M(q^{(1)})\| \leq \frac{3}{4} \|M(q^{(1)})\|. \tag{16}$$

If $\lambda^{(t)} \leq 1200 \lambda^*$,

$$\|M(q^{(t)})\| \leq \frac{10}{9} \lambda^{(t)} \leq \frac{10}{9} \cdot 1200 \lambda^* \leq 1350 \lambda^*. \tag{17}$$

From now on, suppose that $\lambda^{(t)} > \frac{1}{2} \lambda^{(1)}$ and $\lambda^{(t)} > 1200 \lambda^*$ for all $1 \leq t \leq T$. Observe that $\lambda^{(t)} > 1200 \lambda^*$ implies

$$\|M(q^{(t)})\| \geq 1000 \lambda^* \geq 1000 \|M(q^*)\|, \tag{18}$$

where the last step follows from (13). Hence, all $q^{(t)}$ satisfy the requirement in Lemma 16. Thus, we can show that $0 \leq q^{(t+1)} \leq q^{(t)}$ for all $t \in [T]$. Thus, $\|M(q^{(t)})\| \leq \|M(q^{(1)})\|$. An immediate consequence is that for the matrices $\frac{9}{10 \lambda^{(1)}} M(q^{(t)})$ appearing in the definition of $U^{(t)}$ in (5), we have

$$\| \frac{9}{10 \lambda^{(1)}} M(q^{(t)}) \| \leq \|M(q^{(1)})\|^{-1} \|M(q^{(t)})\| \leq 1.$$

This allows us to apply the regret bound of matrix multiplicative weights update method (Allen-Zhu et al., 2015) and obtain

$$\begin{aligned}
&\|\sum_{t=1}^{T} M(q^{(t+1)})\| \\
&\leq \sum_{t=1}^{T}\langle M(q^{(t+1)}), U^{(t)}\rangle + \frac{9}{10\lambda^{(1)}}\sum_{t=1}^{T}\langle M(q^{(t+1)}), U^{(t)}\rangle \|M(q^{(t+1)})\| + \frac{10\lambda^{(1)}}{9}\log d \\
&\leq 2\sum_{t=1}^{T}\langle M(q^{(t+1)}), U^{(t)}\rangle + \frac{11}{9}\|M(q^{(1)})\| \cdot \log d. \qquad (19)
\end{aligned}$$

It remains to upper bound the cross term on the right-hand side and lower bound the left-hand side. To this end, observe that for any iteration t, if $\langle q^{(t)}, \tilde{\beta}^{(t)}\rangle \leq \frac{1}{5}\lambda^{(1)}$, we have $q^{(t+1)} = q^{(t)}$ and therefore

$$\langle M^{(t+1)}, U^{(t)}\rangle = \sum_{i=1}^{n} q_i^{(t)} x_i^\top U^{(t)} x_i \leq \frac{11}{10}\sum_{i=1}^{n} q_i^{(t)}\tilde{\beta}_i^{(t)} \leq \frac{11}{50}\lambda^{(1)} \leq \frac{121}{500}\|M(q^{(1)})\|. \qquad (20)$$

Otherwise, by Lemma 16, we have $q^{(t+1)} \leq q^{(t)}$. Therefore,

$$\langle M(q^{(t)}), U^{(t)}\rangle \leq \langle M(q^{(1)}), U^{(t)}\rangle \leq \|M(q^{(1)})\|. \qquad (21)$$

Lemma 16 also tells that

$$\langle M^{(t+1)}, U^{(t)}\rangle \leq \frac{1}{16}\langle M^{(t)}, U^{(t)}\rangle \overset{(21)}{\leq} \frac{1}{16}\|M(q^{(1)})\|. \qquad (22)$$

Combining (20) and (22) gives

$$\langle M^{(t+1)}, U^{(t)}\rangle \leq \frac{1}{4}\|M(q^{(1)})\|, \text{ for all } 1 \leq t \leq T. \qquad (23)$$

To lower bound the spectral norm of $\sum_{t=1}^{T} M(q^{(t+1)})$, again by the result $0 \leq q^{(t+1)} \leq q^{(t)}$, we have $M(q^{(T+1)}) \preceq M(q^{(t+1)}$ for all $t \leq T$. Hence,

$$\sum_{t=1}^{T} M(q^{(T+1)}) \preceq \sum_{t=1}^{T} M(q^{(t+1)}).$$

This implies

$$T\|M(q^{(T+1)})\| \leq \|\sum_{t=1}^{T} M(q^{(t+1)})\|. \qquad (24)$$

Putting (19), (23) and (24) together, and choosing $T = 8\log d$ gives $\|M(q^{(T+1)})\| \leq \frac{3}{4}\|M(q^{(1)})\|$.

Now we consider the failure probability of the algorithm. The randomness occurs when invoking APPROXEV and $\mathrm{MMWUScore}$. By Lemma 8 and Lemma 7 respectively, each iteration may fail with probability $\frac{\delta''}{T}$. Thus, by union bound over all T iterations, with probability $1 - \delta''$, the algorithm succeeds.

Lastly, we analyze the running time. By Lemma 8, the computational cost of APPROXEV through the T iterations is $O(Tnd\log\frac{T}{\delta''})$, which amount to be $\tilde{O}(nd\log\frac{1}{\delta''})$ as $T = 8\log d$. By Lemma 7, the computational cost of $\mathrm{MMWUScore}$ is $\tilde{O}\big(\sum_{t=1}^{T} tnd\log\frac{T}{\delta''}\big) = \tilde{O}(nd\log\frac{1}{\delta''})$. When $\langle q^{(t)}, \tilde{\beta}^{(t)}\rangle > \frac{1}{5}\lambda^{(1)}$, REFINE will invoke 1D-FILTER, whose running time is $O\big(n\log\frac{\|\beta^{(t)}\|_\infty}{\langle q^{(t)}, \tilde{\beta}^{(t)}\rangle}\big) \leq O\big(n\log\frac{\|\beta^{(t)}\|_\infty}{\lambda^{(1)}}\big)$. Since $\beta_i^{(t)} = x_i^\top U^{(t)} x_i$, $\|x_i\| \leq \gamma$, $\mathrm{tr}(U^{(t)}) = 1$, it is not hard to see that $\beta_i^{(t)} \leq \gamma^2$ for all $t \in [T]$ and $i \in [n]$. Thus, $\|\beta^{(t)}\|_\infty \leq \gamma^2$. To lower bound $\lambda^{(1)}$, note that the condition that we invoke REFINE in REWEIGHT is that this quantity is greater than $1250\lambda^*$. Therefore, the running time of invoking 1D-FILTER once is $O(n\log\frac{\gamma}{\lambda^*})$.

This completes the proof of the theorem. □

A.6 Analysis of OPTIMIZE

Recall the hinge loss:

$$\ell_\tau(w; S, p) = \sum_{i=1}^{n} p_i \cdot \max\left\{0, 1 - \frac{1}{\tau} y_i w \cdot x_i\right\}$$

We consider finding a solution $\hat{v} \in W$ such that

$$\ell_\tau(\hat{v}; S, p) \leq \min_{w \in W} \ell_\tau(w; S, p) + \kappa,$$

where $W := \{w \in \mathbb{R}^d : \|w - u\| \leq r\}$ and κ is an absolute constant.

We need a standard regret bound of online gradient descent, first considered in Zinkevich (2003).

Lemma 17 (Theorem 3.1 in Hazan (2019))**.** *Consider a sequence of samples $\{(x_t, y_t)\}_{t=1}^T$ and convex function $f(w; x, y)$. Let W be the constraint set with diameter $\mathrm{diam}(W)$. Suppose $G > 0$ is such that $\left\|\nabla_w f(w; x_t, y_t)\right\| \leq G$ for all $1 \leq t \leq T$. Then online gradient descent with step sizes $\rho_t = \frac{\mathrm{diam}(W)}{G\sqrt{t}}$ guarantees that for any $w \in W$,*

$$\mathrm{Reg}_T(w) := \sum_{t=1}^{T} f(w_t; x_t, y_t) - \sum_{t=1}^{T} f(w; x_t, y_t) \leq \frac{3}{2} G \cdot \mathrm{diam}(W) \cdot \sqrt{T}.$$

The following result is known as online-to-batch conversion, which is useful to analyze the generalization error of the iterate produced by an online learner.

Lemma 18 (Corollary 2 in Cesa-Bianchi et al. (2004))**.** *Consider the same conditions as Lemma 17. Further assume that the sequence of the samples are independent draws from a distribution D, and $\|f\|_\infty \leq M$. Let $\mathrm{risk}(w) := \mathbb{E}_{(x,y)\sim D}[f(w; x, y)]$ and denote $w^* = \arg\min_{w \in W} \mathrm{risk}(w)$. Denote $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. Then with probability at least $1 - \delta$,*

$$\mathrm{risk}(\bar{w}) \leq \mathrm{risk}(w^*) + \frac{\mathrm{Reg}_T(w^*)}{T} + 2M\sqrt{\frac{2\log(2/\delta)}{T}}.$$

A.7 Proof of Theorem 9

Proof. We think of the algorithm as applying the online gradient descent to the function $f(w; x, y) = \max\left\{0, 1 - \frac{1}{\tau} y w \cdot x\right\}$. First, for any w and w' in W, it is easy to show that $\left\|w - w'\right\| \leq \|w - u\| + \left\|w' - u\right\| \leq 2r$; thus $\mathrm{diam}(W) = 2r$. Second, $\left\|\nabla_w f(w; x, y)\right\| \leq \frac{1}{\tau}\|x\| \leq \frac{\gamma}{\tau}$.

Now Lemma 17 guarantees that with $\rho_t = \frac{2r\tau}{\gamma\sqrt{t}}$, we have for all $w \in W$,

$$\sum_{t=1}^{T} f(v_t; x_{i_t}, y_{i_t}) - \sum_{t=1}^{T} f(w; x_{i_t}, y_{i_t}) \leq \frac{3}{2} \cdot \frac{\gamma}{\tau} \cdot 2r \cdot \sqrt{T} = \frac{3r\gamma}{\tau}\sqrt{T}. \tag{25}$$

Lastly, $\|f\|_\infty \leq 1 + \frac{1}{\tau}|w \cdot x| \leq \frac{2(b+r)\gamma}{\tau}$ where we use the fact $|w \cdot x| \leq |u \cdot x| + \left|(w - u) \cdot x\right| \leq b + \|w - u\| \cdot \|x\| \leq b + r \cdot \gamma \leq (b + r)\gamma$.

Therefore, combining the above with Eq. (25) and Lemma 18, we have with probability $1 - \delta'/2$,

$$\begin{aligned}\ell_\tau(\hat{v}; S, p) &\leq \min_{w \in W} \ell_\tau(w; S, p) + \frac{3r\gamma}{\tau\sqrt{T}} + \frac{2(b+r)\gamma}{\tau} \cdot \sqrt{\frac{2\log(4/\delta')}{T}} \\ &\leq \min_{w \in W} \ell_\tau(w; S, p) + \frac{6(b+r)\gamma}{\tau} \cdot \sqrt{\frac{2\log(4/\delta')}{T}}.\end{aligned}$$

Hence, if we pick

$$T = \frac{1}{\kappa^2} \cdot \frac{72(b+r)^2\gamma^2}{\tau^2} \cdot \log\frac{4}{\delta'}, \tag{26}$$

we obtain

$$\ell_\tau(\hat{v}; S, p) \leq \min_{w \in W} \ell_\tau(w; S, p) + \kappa.$$

Finally, we note that the per-iteration cost of OPTIMIZE is $O(d)$ since both the stochastic update and projection on W run in $O(d)$ time. This completes the proof. □

B USEFUL LEMMAS

Lemma 19 (Proposition 8 of Shen (2021b))**.** *Let S_{C} be a set of i.i.d. instances drawn from $D_{u,b}$. If $|S_{\mathrm{C}}| \geq d \cdot \mathrm{polylog}(d, \frac{1}{b}, \frac{1}{\delta})$, then with probability $1-\delta$, $\|M\| \leq c$ for some absolute constant $c > 0$, where $M := \frac{1}{|S_{\mathrm{C}}|} \sum_{x \in S_{\mathrm{C}}} xx^\top$.*

Lemma 20 (Lemma 11 of Shen (2021b))**.** *Assume $\eta < \frac{1}{2}$. By making a number of $m \geq \Omega\big(\frac{1}{b}(n + \log\frac{1}{\delta})\big)$ calls to $\mathrm{EX}(D, w^*)$, we obtain n samples to form S' with probability $1 - \delta$.*

Lemma 21 (Lemma 12 of Shen (2021b))**.** *Assume $\eta \leq c_5 \epsilon$. If $|S| \geq 24 \ln \frac{1}{\delta}$, then with probability $1-\delta$, $|S_{\mathrm{C}}| \geq \frac{3}{4}|S|$.*

Lemma 22. *Suppose S is a set of i.i.d. instances drawn from $D_{u,b}$. Then with probability $1-\delta$, $\max_{x\in S} \|x\| \leq \sqrt{d} \log\big(\frac{e|S|}{c_8 b \delta}\big)$.*

Proof. By Lemma 26 of Shen (2021b), we have for any $\alpha > 0$,

$$\mathrm{Pr}_{x \sim D_{u,b}}\big(\|x\| \geq \alpha\big) \leq \frac{e}{c_8 b} \exp(-\alpha/\sqrt{d}),$$

implying that

$$\mathrm{Pr}_{x \sim D_{u,b}}\big(\max_{x\in S} \|x\| \geq \alpha\big) \leq |S| \cdot \frac{e}{c_8 b} \exp(-\alpha/\sqrt{d}).$$

Thus, with probability $1-\delta$, we have

$$\max_{x\in S} \|x\| \leq \sqrt{d} \log\left(\frac{e|S|}{c_8 b \delta}\right).$$

The proof is complete. □

Lemma 23 (Theorem 2.2 of Dong et al. (2019))**.** *Assume D' is sub-gaussian with identity covariance matrix. There exists an algorithm that given the corrupted instance set $S = \{z_1, \dots, z_n\}$, runs in time $\tilde{O}(nd)$ and with probability $1-\delta$, outputs $\hat{\mu}$ such that $\|\hat{\mu} - \mu^*\| \leq O\big(\eta\sqrt{\log(1/\eta)} + \sqrt{(d + \log(1/\delta))/n}\big)$.*

Lemma 24 (Theorem 2.4 of Dong et al. (2019))**.** *Let $\alpha \in (0, 1/2)$, $\alpha' \geq 2\alpha$, and let $q = (q_1, \dots, q_n)$ and $\beta = (\beta_1, \dots, \beta_n)$ be non-negative vectors such that $\|q\|_1 \leq 1$. Suppose there exists two disjoint sets $S_{\mathrm{C}} \cup S_{\mathrm{D}} = [n]$, and*

$$\langle q_{S_{\mathrm{C}}}, \beta_{S_{\mathrm{C}}} \rangle \leq \alpha \langle q, \beta \rangle.$$

Then $1\mathrm{D}\text{-}\textsc{Filter}(q, \beta, \alpha')$ runs in time $O\big(n \log \frac{\|\beta\|_\infty}{\alpha'\langle q, \beta\rangle}\big)$ and outputs $0 \leq q' \leq q$ satisfying the following conditions:

1. *more weights are removed from S_{D} than from S_{C}, i.e. $\sum_{i \in S_{\mathrm{C}}} q_i - q_i' \leq \sum_{i\in S_{\mathrm{D}}} q_i - q_i'$;*
2. *$\langle q', \beta \rangle \leq \alpha' \langle q, \beta \rangle$.*

Note that we can run 1D-Filter with $\alpha' = 2\alpha$ which is strictly less than 1; hence the second part of the above lemma guarantees that the new weight vector q' decreases the overall score.

C HYPER-PARAMETER SETTING FOR ALGORITHM 1

For readers interested in the constant factor hidden in the $\Theta(\cdot)$ notation in b_k and other hyper-parameters used in Algorithm 1, we clarify it here.

Our hyper-parameter setting of Algorithm 1 is same as Shen (2021b). In the following, $c_0, c_1, \dots, c_8$ and C_2 were positive absolute constants that were set out in Appendix A of Shen (2021b).

Let $g(t) = c_2\left(2t\exp(-t) + \frac{c_3\pi}{4}\exp\left(-\frac{c_4 t}{4\pi}\right) + 16\exp(-t)\right)$. Let $\bar{c} \geq 8\pi/c_4$ be such that $g(\bar{c}) \leq 2^{-8}\pi$; it is easy to verify its existence. Given such constant $\bar{c}$, we set the constant $\kappa = \exp(-\bar{c})$, $r_1 = 1$ and $r_k = 2^{-k-6}$ for $k \geq 2$, $b_k = \bar{c} \cdot r_k$, $\tau_k = c_0 \kappa \cdot \min\{b_k, 1/9\}$, $\delta_k = \frac{\delta}{(k+1)(k+2)}$, and choose $\xi_k = \min\left\{\frac{1}{2}, \frac{\kappa^2}{16}\left(1 + 4\sqrt{C_2} z_k/\tau_k\right)^{-2}\right\}$ where $z_k = \sqrt{b_k^2 + r_k^2}$. It is easy to see that all ξ_k's are lower bounded by a constant $c_6 := \min\left\{\frac{1}{2}, \frac{\kappa^2}{16}\left(1 + \frac{4}{c_0\kappa\bar{c}}\sqrt{C_2\bar{c}^2 + C_2}\right)^{-2}\right\}$ and are upper bounded by $\frac{1}{2}$. Our theoretical guarantee holds for any noise rate $\eta \leq c_5\epsilon$, where the constant $c_5 := \frac{c_8}{2\pi}\bar{c} c_1 c_6$ (which is the constant c' in Theorem 3).

We set the total number of phases $k_{\max} = \log\left(\frac{\pi}{32 c_1 \epsilon}\right)$. For any phase $k \geq 1$, we set $m_k = \frac{d}{b_k} \cdot \mathrm{polylog}(d, \frac{1}{b_k}, \frac{1}{\delta_k})$ which is the number of calls to $\mathrm{EX}(D, w^*)$ during rejection sampling.