Do Outliers Ruin Collaboration?

Mingda Qiao [1]

Abstract

We consider the problem of learning a binary classifier from n different data sources, among which at most an η fraction are adversarial. The *overhead* is defined as the ratio between the sample complexity of learning in this setting and that of learning the same hypothesis class on a single data distribution. We present an algorithm that achieves an $O(\eta n + \ln n)$ overhead, which is proved to be worst-case optimal. We also discuss the potential challenges to the design of a computationally efficient learning algorithm with a small overhead.

1. Introduction

Consider the following real-world scenario: we would like to train a speech recognition model based on labeled examples collected from different users. For this particular application, a high *average* accuracy over all users is far from satisfactory: a model that is correct on 99.9% of the data may still go seriously wrong for a small yet non-negligible 0.1% fraction of the users. Instead, a more desirable objective would be finding personalized speech recognition solutions that are accurate for *every single user*.

There are two major challenges to achieving this goal, the first being user heterogeneity: a model trained exclusively for users with a particular accent may fail miserably for users from another region. This challenge hints that a successful learning algorithm should be adaptive: more samples need to be collected from users with atypical data distributions. Equally crucial is that a small fraction of the users are malicious (e.g., they are controlled by a competing corporation); these users intend to mislead the speech recognition model into generating inaccurate or even ludicrous outputs.

Motivated by these practical concerns, we propose the *Robust Collaborative Learning* model and study from a theoretical perspective the complexity of learning in the presence of untrusted collaborators. In our model, a learning algorithm interacts with n different users, each associated with a data distribution $\mathcal{D}_i$. As mentioned above, a successful learning algorithm should, ideally, find personalized classifiers $f_1, f_2, \ldots, f_n$ for different distributions, such that

$$\mathrm{err}_{\mathcal{D}_i}(f_i) \triangleq \Pr_{x\sim\mathcal{D}_i}[f_i(x) \neq f^*(x)] < \epsilon$$

holds for every $i \in [n]$, where $f^*(x)$ denotes the true label of sample x. Further complicating the situation is that the algorithm can only interact with the data distributions via the users, each of which is either *truthful* or *adversarial*. A truthful user always provides the learning algorithm with independent samples drawn from his distribution together with the correct labels, whereas the labeled samples collected from adversarial users are arbitrary.

In the presence of malicious users, it is clearly impossible to learn an accurate classifier for every single distribution: an adversary may choose to provide no information about his data distribution. Therefore, a more realistic objective is to satisfy all the truthful users, i.e., to learn n classifiers $f_1, f_2, \ldots, f_n$ such that $\mathrm{err}_{\mathcal{D}_i}(f_i) < \epsilon$ holds for every truthful user i.

Naïvely, one could ignore the prior knowledge that samples from truthful users are labeled by the same function, and run n independent copies of the same learning algorithm for the n users. This straightforward approach clearly needs at least n times as many samples as that required by learning on a single data distribution. Following the terminology of Blum et al. (2017), we say that this naïve algorithm leads to an $\Omega(n)$ sample complexity *overhead*. The notion of overhead measures the extent to which learning benefits from the collaboration and sharing of information among different parties. Blum et al. (2017) proposed a learning algorithm that achieves an $O(\ln n)$ overhead for the case that all users are truthful, i.e., $\eta = 0$. We are then interested in answering the following natural question: can we still achieve a sublinear overhead for the case that $\eta > 0$, at least when η is sufficiently small? In other words, *do adversaries ruin the efficiency of collaboration?*

[1] Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, Beijing, China. Correspondence to: Mingda Qiao <ACMonsterQiao@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

1.1. Model and Preliminaries

Similar to the classic *Probably Approximately Correct (PAC) learning* framework due to Valiant (1984), we consider the binary classification problem on a set $\mathcal{X}$. The hypothesis class $\mathcal{F}$ is a collection of binary functions on $\mathcal{X}$ with VC-dimension d. The elements in $\mathcal{X}$ are labeled by an unknown target function $f^* \in \mathcal{F}$.[1]

Suppose that $\mathcal{D}$ is a probability distribution on set $\mathcal{X}$. Let $\mathcal{O}_{\mathcal{F}}$ denote the oracle that, given a set $S = \{(x_i, y_i)\}$ of labeled examples, either returns a classifier $f \in \mathcal{F}$ that is consistent with the examples (i.e., $f(x_i) = y_i$ for every $(x_i, y_i) \in S$) or returns $\perp$ if $\mathcal{F}$ contains no such consistent classifiers. A classic result in PAC learning states that if

$$m = \Theta\left(\frac{d\ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}\right)$$

independent labeled samples $S = \{(x_i, f^*(x_i)) : i \in [m]\}$ are drawn from $\mathcal{D}$, with probability at least $1 - \delta$, inequality $\mathrm{err}_{\mathcal{D}}(f) < \epsilon$ holds for every possible output $f = \mathcal{O}_{\mathcal{F}}(S)$ (Blumer et al., 1989).

In the Robust Collaborative Learning setting, we consider n different data distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ supported on $\mathcal{X}$. A learning algorithm interacts with these distributions via n *user oracles* $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$, each of which operates in one of two different modes: *truthful* or *adversarial*. Upon each call to a truthful oracle $\mathcal{O}_i$, a sample x is drawn from distribution $\mathcal{D}_i$ and the labeled sample $(x, f^*(x))$ is returned. On the other hand, an adversarial oracle $\mathcal{O}_i$ may output an arbitrary pair in $\mathcal{X} \times \{0, 1\}$ each time.[2]

We define (ϵ, δ, η)-learning in the Robust Collaborative Learning model as the task of learning an ϵ-accurate classifier for each truthful user with probability $1 - \delta$, under the assumption that at most an η fraction of the oracles are adversarial.

Definition 1.1 ((ϵ, δ, η)-learning)**.** *Algorithm $\mathcal{A}$ is an (ϵ, δ, η)-learning algorithm if $\mathcal{A}$, given a concept class $\mathcal{F}$ and access to n user oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$ among which at most ηn oracles are adversarial, outputs functions $f_1, f_2, \ldots, f_n : \mathcal{X} \to \{0, 1\}$, such that with probability at least $1 - \delta$, $\mathrm{err}_{\mathcal{D}_i}(f_i) < \epsilon$ holds simultaneously for every truthful oracle $\mathcal{O}_i$.*

We also formally define the sample complexity of (ϵ, δ, η)-learning.

Definition 1.2 (Sample Complexity)**.** *Let $M_{\mathcal{A}}(\mathcal{F}, \{\mathcal{O}_i\})$ denote the expected number of times that algorithm $\mathcal{A}$ calls oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$ in total, when it runs on hypothesis class $\mathcal{F}$ and user oracles $\{\mathcal{O}_i\}$. The sample complexity of (ϵ, δ, η)-learning a concept class with VC-dimension d from n users is defined as:*

$$m_{n,d}(\epsilon, \delta, \eta) \triangleq \inf_{\mathcal{A}} \sup_{\mathcal{F}, \{\mathcal{O}_i\}} M_{\mathcal{A}}(\mathcal{F}, \{\mathcal{O}_i\}).$$

Here the infimum is over all (ϵ, δ, η)-learning algorithms $\mathcal{A}$. The supremum is taken over all hypothesis classes $\mathcal{F}$ with VC-dimension d and user oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$, among which at most an η fraction are adversarial.

The *overhead* of Robust Collaborative Learning is defined as the ratio between the sample complexity $m_{n,d}(\epsilon, \delta, \eta)$ and its counterpart in the classic PAC learning setting, $m_{1,d}(\epsilon, \delta, 0)$. To simplify the notations and restrict our attention to the dependence of overhead on parameters n, d and η, we assume that $\epsilon = \delta = 0.1$ in our definition of overhead.[3]

Definition 1.3 (Overhead)**.** *For $n, d \in \mathbb{N}$ and $\eta \in [0, 1]$, the sample complexity overhead of Robust Collaborative Learning is defined as*

$$o(n, d, \eta) \triangleq \frac{m_{n,d}(\epsilon, \delta, \eta)}{m_{1,d}(\epsilon, \delta, 0)},$$

where $\epsilon = \delta = 0.1$.

Following our definition of the overhead, the results in (Blum et al., 2017) imply that when all users are truthful (i.e., when $\eta = 0$) and $n = O(d)$, $o(n, d, 0) = O(\ln n)$. They also proved the tightness of this bound in the special case that $n = \Theta(d)$.

1.2. Our Results

Information-theoretically, collaboration can be robust. In Section 3, we present our main positive result: a learning algorithm that achieves an $O(\eta n + \ln n)$ sample complexity overhead when $n = O(d)$. Our result recovers the $O(\ln n)$ overhead upper bound due to Blum et al. (2017) for the special case $\eta = 0$. In Section 4, we complement our positive result with a lower bound, which states that an $\Omega(\eta n)$ overhead is inevitable in the worst case. In light of the previous $\Omega(\ln n)$ overhead lower bound for the special case that $n = \Theta(d)$ (Blum et al., 2017), our learning algorithm achieves an optimal overhead when parameters n and d differ by a bounded constant factor.

Our characterization of the sample complexity in Robust Collaborative Learning indicates that efficient cooperation is possible even if a small fraction of arbitrary outliers are present. Moreover, the overhead is largely determined by ηn, the maximum possible number of adversaries. Our

[1] This is known as the *realizable* setting of PAC learning.

[2] Our results hold even if the adversarial oracles are allowed to collude and they know the samples previously drawn by truthful oracles.

[3] This definition only changes by a constant factor when 0.1 is replaced by other sufficiently small constants.

results suggest that for practical applications, the learning algorithm could greatly benefit from a relatively clean pool of data sources.

Computationally, outliers may ruin collaboration. Our study focuses on the sample complexity of Robust Collaborative Learning, yet also important in practice is the amount of computational power required by the learning task. Indeed, the algorithm that we propose in Section 3 is inefficient due to an exhaustive enumeration of the set of truthful users, which takes exponential time. In Section 5, we provide evidence that hints at a time-sample complexity tradeoff in Robust Collaborative Learning. Informally, we conjecture that any learning algorithm with a sublinear overhead must run in super-polynomial time. In other words, while the presence of adversaries does not seriously increase the sample complexity of learning, it may still ruin the efficiency of collaboration by significantly increasing the computational burden of this learning task. We support our conjecture with known hardness results in computational complexity theory.

2. Related Work

Most related to our work is the recent *Collaborative PAC Learning* model proposed by Blum et al. (2017). They also considered the task of learning the same binary classifier on different data distributions, yet all users are assumed to be truthful in their model. In fact, the Robust Collaborative Learning model reduces to the *personalized setting* of their model when $\eta = 0$. Here the word "personalized" emphasizes the assumption that each user may receive a specialized classifier tailored to his distribution.

In addition to the personalized setting, they also studied the *centralized setting*, in which all the n users should receive the same classifier. They proved that a poly-logarithmic overhead is still achievable in this more challenging setting. In our Robust Collaborative Learning model, however, centralized learning is in general impossible due to the indistinguishability between truthful and adversarial users. The following simple impossibility result holds for extremely simple concept classes and even when infinitely many samples are available.

Proposition 2.1. *For any $\epsilon \in [0, 1)$, $\delta \in \left[0, \frac{1}{2}\right)$ and $\eta \in (0, 1]$, no algorithms (ϵ, δ, η)-learn any concept class of VC-dimension $d \ge 2$, under the restriction that all users should receive the same classifier.*

Proof of Proposition 2.1. Let x_0 and x_1 be two different samples that can be shattered by $\mathcal{F}$. Choose $f_0, f_1 \in \mathcal{F}$ such that $f_0(x_0) = f_0(x_1) = f_1(x_0) = 0$ and $f_1(x_1) = 1$. Let n be large enough such that $\eta n \ge 1$. Construct (degenerate) distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ such that $\mathcal{D}_1(x_1) = \mathcal{D}_2(x_1) = 1$ and $\mathcal{D}_i(x_0) = 1$ for each $3 \le i \le n$.

Consider the following two cases:

- The target function is f_0. The only adversarial user, $\mathcal{O}_1$, misleads the learning algorithm by outputing the labeled example $(x_1, 1)$.
- The target function is f_1. The only adversarial user, $\mathcal{O}_2$, misleads the learning algorithm by outputing the labeled example $(x_1, 0)$.

Note that in both cases, oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ always return $(x_1, 1)$ and $(x_1, 0)$ respectively, while all other oracles return $(x_0, 0)$. Consequently, no algorithms can distinguish these two cases with success probability strictly greater than $\frac{1}{2}$. Thus, any learning algorithm would have a failure probability of at least $\frac{1}{2} > \delta$. □

A related line of research is multi-task learning (Caruana, 1997; Baxter, 2000; Ben-David et al., 2002; 2003), which studies the problem of learning multiple related tasks simultaneously with significantly fewer samples. Most work in this direction assumes certain relation (e.g., a transfer function) between the given learning tasks. In contrast to multi-task learning, our work focuses on the problem of learning the same classifier on multiple data distributions, without assuming any similarity between these underlying distributions.

Also relevant to our study is the work on robust statistics, i.e., the study of learning and estimation in the presence of noisy data and arbitrary outliers; see Lai et al. (2016); Charikar et al. (2017); Diakonikolas et al. (2016; 2017; 2018) and the references therein for some recent work in this line of research. Classic problems in this regime include the estimation of the mean and covariance of a high-dimensional distribution, given a dataset consisting of samples drawn from the distribution and a small fraction of arbitrary outliers. Our model differs from this line of research in that we consider a general classification setting, and the learning algorithm is allowed to sample different sources adaptively, instead of learning from a given dataset of fixed size.

3. An Iterative Learning Algorithm

In this section, we present an iterative (ϵ, δ, η)-learning algorithm achieves an $O(\eta n + \ln n)$ overhead when $n = O(d)$. Here n is the number of users, and d denotes the VC-dimension of the hypothesis class $\mathcal{F}$. Since $\mathcal{F}$ can be large and even infinite, we assume that the algorithm access $\mathcal{F}$ via an oracle $\mathcal{O}_{\mathcal{F}}$ that, given a set $S = \{(x_i, y_i)\}$ of labeled examples, either returns a classifier $f \in \mathcal{F}$ such that $f(x_i) = y_i$ holds for each pair $(x_i, y_i) \in S$, or returns $\perp$ if $\mathcal{F}$ does not contain any consistent functions. The

Algorithm 1 Iterative Robust Collaborative Learning

```
Input: Parameters $n$, $d$, $\epsilon$, $\delta$, $\eta$.
Output: Classifiers $f_1, f_2, \ldots, f_n$.
$r \leftarrow 1$; $G_1 \leftarrow [n]$;
while $\lfloor \eta n \rfloor \le \frac{|G_r|}{10}$ do
    $\delta_r \leftarrow \frac{\delta}{5r^2}$;
    $\hat{f}_r \leftarrow \mathsf{Candidate}(G_r, d, \epsilon, \delta_r)$;
    $G_{r+1} \leftarrow \mathsf{Test}(G_r, \hat{f}_r, \epsilon, \delta_r)$;
    Set $f_i \leftarrow \hat{f}_r$ for each $i \in G_r \setminus G_{r+1}$;
    $r \leftarrow r + 1$;
end while
for $i \in G_r$ do
    $S_i \leftarrow \Theta\left(\frac{d\ln(1/\epsilon)+\ln(n/\delta)}{\epsilon}\right)$ labeled samples from $\mathcal{O}_i$;
    $f_i \leftarrow \mathcal{O}_{\mathcal{F}}(S_i)$;
end for
Output $f_1, f_2, \ldots, f_n$;
```

algorithm interacts with the underlying data distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ via n example oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$, among which at most an η fraction are adversarial.

3.1. Algorithm

Our algorithm is formally described in Algorithms 1 through 3. The main algorithm proceeds in rounds and maintains a set G_r of the indices of *active users* at the beginning of round r, i.e., users who have not received an ϵ-accurate classifier so far. When $\lfloor \eta n \rfloor$, the maximum possible number of adversaries, is below $\frac{|G_r|}{10}$, the algorithm invokes subroutine Candidate to find a *candidate classifier* $\hat{f}_r$. Then, Algorithm 1 calls the validation procedure Test to check whether $\hat{f}_r$ is accurate for each user $i \in G_r$ (with respect to accuracy threshold ϵ). If so, the algorithm marks the output for user i as $\hat{f}_r$; otherwise, user i stays in set G_{r+1} for the next round. When the proportion of adversaries reaches $\frac{1}{10}$, the algorithm learns for the remaining users independently: for each active user, it draws samples from his oracle and outputs an arbitrary classifier that is consistent with his data.

3.2. Analysis of Subroutines

Subroutine Candidate (Algorithm 2) is the key to the sample efficiency of our algorithm, as it enables us to learn a candidate classifier that is accurate simultaneously for a constant fraction of the active users, using only a nearly-linear number of samples (with respect to parameters $|G|$ and d). Subroutine Test (Algorithm 3) further checks whether the learned classifier is accurate enough for each active user. This allows us to determine whether a user should remain active in the next iteration. We devote this subsection to the analysis of these two subroutines.

Algorithm 2 $\mathsf{Candidate}(G, d, \epsilon, \delta)$

```
Input: Index set $G$, parameters $d$, $\epsilon$ and $\delta$.
Output: Candidate classifier $\hat{f} \in \mathcal{F}$.
$M \leftarrow \Theta\left(\frac{d\ln(1/\epsilon)+\ln\left(2^{|G|}/\delta\right)}{\epsilon} + |G|\ln\frac{|G|}{\delta}\right)$;
for $i \in G$ do
    $S_i \leftarrow \frac{4M}{|G|}$ labeled samples from $\mathcal{O}_i$;
end for
$\mathcal{G} \leftarrow \left\{H \subseteq G : |H| \ge \frac{9}{10}|G|\right\}$;
for $H \in \mathcal{G}$ do
    $\hat{f}_H \leftarrow \mathcal{O}_{\mathcal{F}}(\bigcup_{i \in H} S_i)$;
    if $\hat{f}_H \ne \perp$ then
        Output $\hat{f}_H$;
    end if
end for
```

Algorithm 3 $\mathsf{Test}(G, \hat{f}, \epsilon, \delta)$

```
Input: Set $G$ of indices, candidate function $\hat{f}$, parameters $\epsilon$ and $\delta$.
Output: Set $G'$ of surviving indices.
for $i \in G$ do
    $S_i \leftarrow \Theta\left(\frac{\ln(|G|/\delta)}{\epsilon}\right)$ samples from $\mathcal{O}_i$;
    $\theta_i \leftarrow \frac{1}{|S_i|}\sum_{(x,y) \in S_i} \mathbb{I}\left[\hat{f}(x) \ne y\right]$;
end for
Output $G' = \{i \in G : \theta_i > \frac{3}{4}\epsilon\}$;
```

Lemma 3.1. *Suppose $G \subseteq [n]$ denotes the indices of $|G|$ users, among which at most $\frac{|G|}{10}$ are adversarial. Let $\hat{f}$ denote the output of* $\mathsf{Candidate}(G, d, \epsilon, \delta)$*. With probability $1 - \delta$, the following two conditions hold simultaneously for at least $\frac{|G|}{2}$ indices $i \in G$: (1) $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) \le \frac{\epsilon}{2}$; (2) oracle $\mathcal{O}_i$ is truthful.*

The proof of Lemma 3.1 relies on the following technical claim, which enables us to relate the union of several equal-size datasets to the samples drawn from the uniform mixture of the corresponding distributions.

Claim 3.2. *Suppose $m = \Omega\left(n \ln \frac{n}{\delta}\right)$ balls are thrown into n bins independently and uniformly at random. Then with probability $1 - \delta$, no bins contain more than $\frac{2m}{n}$ balls.*

Proof of Claim 3.2. Let random variable X denote the number of balls in a fixed bin, so $\frac{X}{m}$ is the average of m i.i.d. Bernoulli random variables with mean $\frac{1}{n}$. The Chernoff bound implies that

$$\Pr\left[\frac{X}{m} \ge \frac{2}{n}\right] \le e^{-mD\left(\frac{2}{n}\|\frac{1}{n}\right)} = e^{-\Omega\left(n\ln\frac{n}{\delta}\right)\cdot\Omega\left(\frac{1}{n}\right)} \le \frac{\delta}{n},$$

where the last step holds if we choose a sufficiently large hidden constant in $m = \Omega\left(n \ln \frac{n}{\delta}\right)$. The claim follows from a union bound over the n bins. □

Proof of Lemma 3.1. Let G' denote the indices of truthful users in G. By assumption, $|G'| \ge \frac{9}{10}|G|$ and $\mathcal{F}$ contains a function f^* that is consistent with $\bigcup_{i \in G'} S_i$. This guarantees that Algorithm 2 should return $\hat{f}_H$ as the output when $H = G'$, so function $\mathsf{Candidate}$ is well-defined.

Recall that in Algorithm 2, we set

$$M = \Theta\left(\frac{d\ln(1/\epsilon) + \ln\left(2^{|G|}/\delta\right)}{\epsilon} + |G|\ln\frac{|G|}{\delta}\right).$$

Consider the following thought experiment. For each non-empty $H \subseteq G$, we draw a sequence A_H of M integers, each of which is chosen uniformly and independently at random from H. We also draw M samples from oracle $\mathcal{O}_i$ for each $i \in G$. If all users in H are truthful, the samples together with A_H naturally specify a realization of drawing M samples from the uniform mixture distribution $\mathcal{D}_H \triangleq \frac{1}{|H|}\sum_{i \in H}\mathcal{D}_i$: we arrange the M samples drawn from each distribution into a queue, and when we would like to draw the i-th sample, we simply take the sample at the front of queue $A_H(i)$.

For a fixed non-empty subset $H \subseteq G$ that only contains truthful users, the VC theorem implies that with probability $1 - \frac{\delta}{2^{|G|+1}}$ (over the randomness in both the samples and the choice of A_H), when we draw samples from the uniform mixture $\mathcal{D}_H$ as described above, any function $f \in \mathcal{F}$ that is consistent with the labeled samples satisfies $\mathrm{err}_{\mathcal{D}_H}(f) \le \frac{\epsilon}{10}$. By a union bound over $\le 2^{|G|}$ different sets $H \subseteq G$, the above holds for *every* $H \subseteq G$ simultaneously with probability $1 - 2^{|G|} \cdot \frac{\delta}{2^{|G|+1}} = 1 - \frac{\delta}{2}$.

Recall that in Algorithm 2, we first query each oracle $\mathcal{O}_i$ to obtain a "training set" S_i of size $\frac{4M}{|G|}$ for each $i \in G$. Then we find set $H \subseteq G$ and classifier $\hat{f}_H \in \mathcal{F}$ such that: (1) $|H| \ge \frac{9}{10}|G|$; (2) $\hat{f}_H$ is consistent with all labeled samples in $\bigcup_{i \in H} S_i$. Suppose that H is the set associated with the output of Algorithm 2, and let $H' = \{i \in H : \mathcal{O}_i \text{ is truthful}\}$. Note that $|H'| \ge |H| - \frac{|G|}{10} \ge \frac{4}{5}|G|$.

The crucial observation is that since

$$M = \Omega\left(|G|\ln\frac{|G|}{\delta}\right),$$

Claim 3.2 implies that with probability at least $1 - \frac{\delta}{2}$, each index $i \in H'$ appears less than $\frac{2M}{|H'|} \le \frac{4M}{|G|}$ times in $A_{H'}$. In other words, $\bigcup_{i \in H'} S_i$ is a superset of the M samples that are supposed to be drawn from $\mathcal{D}_{H'}$ (in our thought experiment). Since $\hat{f}_H$ is consistent with $\bigcup_{i \in H'} S_i$, a union bound shows that with probability $1 - 2 \cdot \frac{\delta}{2} = 1 - \delta$, we have

$$\frac{1}{|H'|}\sum_{i \in H'}\mathrm{err}_{\mathcal{D}_i}(\hat{f}_H) = \mathrm{err}_{\mathcal{D}_{H'}}(\hat{f}_H) \le \frac{\epsilon}{10}.$$

This further implies that $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_H) \le \frac{\epsilon}{2}$ holds for at least $\frac{|G|}{2}$ indices $i \in H'$; otherwise, we would have

$$\begin{aligned}\frac{1}{|H'|}\sum_{i \in H'}\mathrm{err}_{\mathcal{D}_i}(\hat{f}_H) &\ge \frac{1}{|H'|}\left(|H'| - \frac{|G|}{2}\right)\cdot\frac{\epsilon}{2}\\ &\ge \left(1 - \frac{5}{8}\right)\cdot\frac{\epsilon}{2} > \frac{\epsilon}{10},\end{aligned}$$

which leads to a contradiction. Here the second step applies $|H'| \ge \frac{4}{5}|G|$. This proves the lemma. □

The following lemma, which directly follows from a Chernoff bound and a union bound, states that with probability $1 - \delta$, $\mathsf{Test}(G, \hat{f}, \epsilon, \delta)$ correctly determines whether $\hat{f}$ has an $O(\epsilon)$ error for each user in G.

Lemma 3.3. *Let G' denote the output of* $\mathsf{Test}(G, \hat{f}, \epsilon, \delta)$. *With probability $1 - \delta$, the following holds for every $i \in G$ simultaneously: (1) if $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) > \epsilon$, $i \in G'$; (2) if $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) \le \frac{\epsilon}{2}$, $i \notin G'$.*

Proof of Lemma 3.3. Fix a truthful oracle $\mathcal{O}_i$ with $i \in G$. Recall that Algorithm 3 sets

$$\theta_i = \frac{1}{|S_i|}\sum_{(x,y) \in S_i}\mathbb{I}\left[\hat{f}(x) \ne y\right].$$

Note that θ_i is the average of $\Omega\left(\frac{\ln(|G|/\delta)}{\epsilon}\right)$ independent Bernoulli random variables, each with mean $\mathrm{err}_{\mathcal{D}_i}(\hat{f})$. Thus, the Chernoff bound implies that with probability $1 - \frac{\delta}{|G|}$, the following two conditions holds simultaneously: (1) if $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) > \epsilon$, $\theta_i > \frac{3}{4}\epsilon$; (2) if $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) \le \frac{\epsilon}{2}$, $\theta_i \le \frac{3}{4}\epsilon$. The lemma follows from a union bound over all $i \in G$. □

3.3. Correctness and Sample Complexity

Now we are ready to prove our main result.

Theorem 3.4. *For any $\epsilon, \delta \in (0, 1]$ and $\eta \in [0, 1]$, Algorithm 1 is an (ϵ, δ, η)-learning algorithm and takes at most*

$$O\left(\frac{d\ln(1/\epsilon)}{\epsilon}(\eta n + \ln n) + \frac{n\ln(n/\delta)}{\epsilon}\right)$$

samples.

By Theorem 3.4, the sample complexity $m_{n,d}(\epsilon, \delta, \eta)$ reduces to $O\left(d(\eta n + \ln n)\right)$ when ϵ and δ are constants and $n \le C \cdot d$ for some constant C. Therefore, when $n = O(d)$, we have the following overhead upper bound:

$$o(n, d, \eta) = \frac{O\left(d(\eta n + \ln n)\right)}{\Theta(d)} = O(\eta n + \ln n).$$

Proof of Theorem 3.4. The proof proceeds by applying Lemmas 3.1 and 3.3 iteratively. In each round r, Lemma 3.1

guarantees that with probability $1-\delta_r$, the learned classifier $\hat{f}_r$ has an error below $\frac{\epsilon}{2}$ for at least $\frac{|G_r|}{2}$ truthful users. By Lemma 3.3, for each such distribution, the "validation error" θ_i should be below $\frac{3}{4}\epsilon$, so these users will exit the algorithm by receiving $\hat{f}_r$ as the classifier, and the number of active users decreases by a factor of $\frac{1}{2}$. Therefore, the while-loop in Algorithm 1 terminates after at most $\lfloor \log_2 n \rfloor + 1$ iterations. Finally, the algorithm satisfies the remaining active users by drawing $\Theta\left(\frac{d\ln(1/\epsilon)+\ln(n/\delta)}{\epsilon}\right)$ samples from each of them. Thus, the VC theorem guarantees that for each truthful user, the learned classifier is ϵ-accurate with probability at least $1-\frac{\delta}{3n}$. By a union bound, with probability at least

$$1-\sum_{r=1}^{\infty} 2\delta_r - n\cdot\frac{\delta}{3n} = 1-\left(\frac{1}{3}+\sum_{r=1}^{\infty}\frac{2}{5r^2}\right)\delta \geq 1-\delta,$$

Algorithm 1 returns an ϵ-accurate classifier for each truthful user.

It remains to bound the sample complexity of Algorithm 1. In round r, the number of active users is at most $|G_r| \leq \frac{n}{2^{r-1}}$. Recall that $\delta_r = \frac{\delta}{5r^2}$. The number of samples that Candidate and Test draw in round r is then upper bounded by

$$\begin{aligned}&O\left(\frac{d\ln(1/\epsilon)+|G_r|\ln(|G_r|/\delta_r)}{\epsilon}\right)\\ =&O\left(\frac{d\ln(1/\epsilon)}{\epsilon}+\frac{n\ln(n/\delta)}{2^r\epsilon}\right).\end{aligned}$$

Therefore, the number of samples drawn in the $O(\ln n)$ iterations is upper bounded by:

$$\begin{aligned}&\sum_{r=0}^{\lfloor \log_2 n\rfloor+1} O\left(\frac{d\ln(1/\epsilon)}{\epsilon}+\frac{n\ln(n/\delta)}{2^r\epsilon}\right)\\ =&O\left(\frac{d\ln(1/\epsilon)\ln n+n\ln(n/\delta)}{\epsilon}\right).\end{aligned} \tag{1}$$

When the while-loop in Algorithm 1 terminates, it holds that $|G_r| \leq 10\eta n = O(\eta n)$. After that, we learn on the remaining distributions separately, using

$$O\left(\eta n\cdot\frac{d\ln(1/\epsilon)+\ln(n/\delta)}{\epsilon}\right) \tag{2}$$

samples in total. Adding (1) and (2) gives the desired sample complexity upper bound. □

4. Overhead Lower Bound

In this section, we show that an $\Omega(\eta n+\ln n)$ overhead is unavoidable when $n=\Theta(d)$. Therefore, the overhead achieved by Algorithm 1 is optimal up to a constant factor, when the number of users is commensurate with the complexity of the hypothesis class. Formally, we have the following theorem.

Theorem 4.1. *For any $n, d \in \mathbb{N}$, $\epsilon \in \left(0, \frac{1}{2}\right]$ and $\delta, \eta \in (0,1)$,*

$$m_{n,d}(\epsilon,\delta,\eta) = \Omega\left(\frac{\eta n d}{\epsilon}\right).$$

Theorem 4.1 directly implies the following lower bound on the overhead:

$$o(n,d,\eta) = \frac{\Omega(\eta n d)}{\Theta(d)} = \Omega(\eta n).$$

Combining this with the previous lower bound $o(n,d,\eta) = \Omega(\ln n)$ when $n=\Theta(d)$ and $\eta = 0$ (Blum et al., 2017)[4], we obtain the desired worst-case lower bound of $\Omega(\eta n+\ln n)$.

Proof of Theorem 4.1. Assume without loss of generality that ηn is an integer between 1 and $n-1$. We consider the binary classification problem on set $\mathcal{X}=[d]\cup\{\perp\}$, while the hypothesis class $\mathcal{F}$ contains all the 2^d binary functions on $\mathcal{X}$ that map $\perp$ to 0. The target function f^* is chosen uniformly at random from $\mathcal{F}$.

Suppose that for $(1-\eta)n-1$ truthful users, the data distribution is the degenerate distribution on $\{\perp\}$, so these truthful users provide no information on the correct classifier f^*. On the other hand, the data distribution of the only remaining truthful user i^* satisfies $\mathcal{D}_{i^*}(x) = \frac{2\epsilon}{d}$ for any $x \in [d]$ and $\mathcal{D}_{i^*}(\perp) = 1-2\epsilon$. By construction, a learning algorithm must draw $\Omega\left(\frac{d}{\epsilon}\right)$ samples from $\mathcal{D}_{i^*}$ in order to learn an ϵ-accurate classifier with a non-trivial success probability $1-\delta$.

Now suppose that each of the ηn adversarial users tries to pretend that he is the truthful user i^*. More specifically, each adversarial user i chooses a function $\tilde{f}_i \in \mathcal{F}$ uniformly at random, and answer the queries as if he is the truthful user with a different target function $\tilde{f}_i$. In other words, upon each request from the learning algorithm, oracle $\mathcal{O}_i$ draws x from $\mathcal{D}_{i^*}$ and returns $\left(x, \tilde{f}_i(x)\right)$.

Recall that the actual target function f^* is also uniformly distributed in $\mathcal{F}$, so from the perspective of the learning algorithm, the truthful user i^* is indistinguishable from the other ηn adversarial users. Therefore, an (ϵ,δ,η)-learning algorithm must guarantee that each of these $(\eta n+1)$ users receives an ϵ-accurate classifier with probability at least $1-\delta$. The sample complexity lower bound from PAC learning theory implies that we must draw $\Omega\left(\frac{d}{\epsilon}\right)$ samples from each such user and thus

$$(\eta n+1)\cdot\Omega\left(\frac{d}{\epsilon}\right) = \Omega\left(\frac{\eta n d}{\epsilon}\right)$$

[4] They proved an $\Omega(\ln n)$ lower bound for the special case that $n=d$, yet their proof directly implies the same lower bound when $n=\Theta(d)$.

samples in total. □

5. Discussion: A Computationally Efficient Algorithm?

Although Algorithm 1 is proved to achieve an optimal sample complexity overhead in certain cases, the algorithm is computationally inefficient and of limited practical use when there are a large number of users. In particular, subroutine Candidate performs an exhaustive search over all user subsets of size $\ge \frac{9}{10}|G|$, and thus may potentially call oracle $\mathcal{O}_{\mathcal{F}}$ exponentially many times. In contrast, the naïve approach that learns for different users separately, though obtaining an $\Omega(n)$ overhead, only makes n calls to oracle $\mathcal{O}_{\mathcal{F}}$. Naturally, one may wonder whether we can achieve the best of both worlds by finding a computationally efficient learning algorithm with a small overhead? We conjecture that such an algorithm, unfortunately, does not exist.

Conjecture 5.1. *For any $\alpha > -1$ and $\beta < 1$, no learning algorithms that make polynomially many calls to oracle $\mathcal{O}_{\mathcal{F}}$ achieve an $O(n^\beta)$ overhead when $\eta = \Omega(n^\alpha)$.*

In words, when there is a non-trivial number of adversaries, any efficient learning algorithm would incur a nearly-linear overhead. We remark that it is necessary to assume $\alpha > -1$ since when ηn, the maximum possible number of adversaries, is a constant, the learning algorithm can enumerate the subset of adversarial users in polynomial time, thus achieving an optimal overhead efficiently. Proving or refuting Conjecture 5.1 would greatly further our understanding of the impact of arbitrary outliers on collaborative learning.

The key to our sample-efficient learning algorithm is that subroutine Candidate identifies a large user group such that some classifier $\hat{f} \in \mathcal{F}$ is consistent with all their labeled samples. Lemma 3.1 further guarantees that $\hat{f}$ is ϵ-accurate for at least half of the users. This allows us to satisfy almost all the users in $O(\ln n)$ iterations, resulting in the $\ln n$ term in the overhead.

We note that finding a group of users with consistent datasets generalizes the problem of finding a large clique in a graph: For an undirected graph with vertices labeled from 1 to n, we construct the user oracles $\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_n$ such that $\mathcal{O}_i$ and $\mathcal{O}_j$ produce conflicting labels on the same data if the edge (i, j) is absent from the graph. Then a group of users have consistent datasets if and only if they form a clique in the corresponding graph.

Unfortunately, Zuckerman (2006) proved that even if the graph is known to contain a hidden clique of size $\Omega(n)$[5], it is still NP-hard to find a clique of size $\Omega(n^{1-\beta})$ for any $\beta < 1$. This indicates that, following the approach of Algorithm 1, a computationally efficient algorithm can only find accurate classifiers for at most $O(n^{1-\beta})$ users in each iteration. As a result, $\Omega(n^\beta)$ iterations would be necessary to satisfy all the n users. The algorithm consequently incurs an $\Omega(n^\beta)$ overhead.

[5] Analogously, in our setting, we know that a large fraction of users have non-conflicting datasets.

References

Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research (JAIR)*, 12(1):149–198, 2000.

Ben-David, S., Gehrke, J., and Schuller, R. A theoretical framework for learning from a pool of disparate data sources. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 443–449, 2002.

Ben-David, S., Schuller, R., et al. Exploiting task relatedness for multiple task learning. *Lecture Notes in Computer Science (LNCS)*, pp. 567–580, 2003.

Blum, A., Haghtalab, N., Procaccia, A. D., and Qiao, M. Collaborative pac learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2389–2398, 2017.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

Caruana, R. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997.

Charikar, M., Steinhardt, J., and Valiant, G. Learning from untrusted data. In *Symposium on Theory of Computing (STOC)*, pp. 47–60, 2017.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS)*, pp. 655–664, 2016.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning (ICML)*, pp. 999–1008, 2017.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robustly learning a gaussian: Getting optimal error, efficiently. In *Symposium on Discrete Algorithms (SODA)*, pp. 2683–2702, 2018.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS)*, pp. 665–674, 2016.

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

Zuckerman, D. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Symposium on Theory of Computing (STOC)*, pp. 681–690, 2006.