Communication-Aware Collaborative Learning

Avrim Blum,[1] **Shelby Heinecke,**[2] **Lev Reyzin**[3]
[1] Toyota Technological Institute at Chicago
[2] Salesforce Research
[3] University of Illinois at Chicago
avrim@ttic.edu, shelby.heinecke@salesforce.com, lreyzin@uic.edu

Abstract

Algorithms for noiseless collaborative PAC learning have been analyzed and optimized in recent years with respect to sample complexity. In this paper, we study collaborative PAC learning with the goal of reducing communication cost at essentially no penalty to the sample complexity. We develop communication efficient collaborative PAC learning algorithms using distributed boosting. We then consider the communication cost of collaborative learning in the presence of classification noise. As an intermediate step, we show how collaborative PAC learning algorithms can be adapted to handle classification noise. With this insight, we develop communication efficient algorithms for collaborative PAC learning robust to classification noise.

Introduction

Collaborative learning was recently formalized by Blum et al. (2017) as a PAC learning model. In this collaborative PAC setting, there is a domain X, over which are k distributions, referred to as *players*. There is also a center node that orchestrates the learning process. The goal of collaborative PAC learning is to learn classifiers from data provided by the players that generalize well on each of players' distributions simultaneously. Note that this is distinct from the related distributed learning setting, where the goal is to learn classifiers that generalize well on the mixture of players' distributions (Balcan et al. 2012).

There are generally a few styles of collaborative PAC learning. In the *personalized learning setting*, which is the main focus of our paper, the goal is to learn a classifier for each player with generalization error less than ϵ, with probability $1 - \delta$. Another setting is the centralized learning setting, where the goal is learn a single classifier with generalization error less than ϵ on each players' distribution with probability $1 - \delta$. The efficiency of a collaborative learning algorithm is assessed by its *overhead*, defined as the ratio of the sample complexity of learning in the collaborative setting to the sample complexity of learning in the single player setting. An overhead of at least k indicates that the collaborative learning algorithm offers no sample complexity benefit over individual PAC learning. An overhead less than k indicates that the collaborative algorithm is more sample efficient than individual PAC learning. Collaborative PAC learning algorithms have been optimized in subsequent works with respect to overhead, and hence sample complexity (Blum et al. 2017; Chen, Zhang, and Zhou 2018; Nguyen and Zakynthinou 2018; Qiao 2018).

Certain difficulties may arise in real-world applications of collaborative PAC learning. First, communicating data between players and the center can be costly. Second, the data from players may be noisy. Consider the example described in (Blum et al. 2017) where k players represent hospitals serving different demographics of the population. In this network of hospitals, each of which generates an abundance of data, transmitting data to the center is costly and thus hospitals want to minimize the amount of data transmitted. Additionally, mistakes may be present in the labels of the data at the hospitals, due to clerical errors and misdiagnoses, among other reasons. Given access to only the noisy data from the hospitals, we wish to learn classifiers that generalize well with respect to each hospital's underlying noiseless distribution. We tackle both difficulties in this paper. First, we develop communication-aware collaborative learning algorithms in the noiseless setting that enjoy reduced communication costs at no penalty to the sample complexity. Then, we develop communication-aware collaborative learning algorithms in the presence of classification noise, where each player has label noise rate $\eta_i < \frac{1}{2}$.

The algorithms and analysis in this work focus on personalized learning. We discuss the applications of our insights and analyses to the centralized learning setting in the Appendix. Omitted proofs are also included in the Appendix.

Previous Work

Algorithms for collaborative PAC learning have been analyzed and optimized in (Blum et al. 2017; Chen, Zhang, and Zhou 2018; Nguyen and Zakynthinou 2018; Qiao 2018) with respect to sample complexity. The collaborative PAC framework was formalized in (Blum et al. 2017), where they also develop an optimal algorithm in the personalized setting with $O(\ln(k))$ overhead and a suboptimal algorithm in the centralized setting with $O(\ln^2(k))$ overhead. We recall their algorithm, which we refer to as Personalized Learning (Algorithm 1), and the corresponding sample complexity result below.

Algorithm 1: Personalized Learning (Blum et al. 2017)

Input: H, k distributions $D_i \sim X$, $\delta' = \delta/2\log(k)$, $\epsilon > 0$
Output: $f_1, ..., f_k \in H$
Let $N_1 = \{1, ..., k\}$;
for $j = 1, ..., \lceil \log(k) \rceil$ **do**
  Draw sample S of size $m_{\epsilon/4,\delta'}$ from mixture $D_{N_j} = \frac{1}{|N_j|}\sum_{i \in N_j} D_i$;
  Select consistent hypothesis $h_j \in H$ on S;
  $G_j \leftarrow \texttt{TEST}(h_j, N_j, \epsilon, \delta')$;
  $N_{j+1} = N_j \setminus G_j$;
  for $i \in G_j$ **do**
    $f_i \leftarrow h_j$;
  end
end
return $f_1, ..., f_k$
Procedure $\texttt{TEST}(h, N, \epsilon, \delta)$
  for $i \in N$ **do**
    Draw $T_i = O\left(\frac{\ln(\frac{|N|}{\epsilon\delta})}{\epsilon}\right)$ samples from D_i;
  end
  return $\{i \mid err_{T_i}(h) \leq \frac{3\epsilon}{4}\}$

Theorem 1 (Blum et al. 2017)**.** *For any $\epsilon, \delta > 0$, and hypothesis class H of finite VC-dimension d, the sample complexity of Personalized Learning is*

$$m = O\left(\frac{\ln(k)}{\epsilon}\left((d+k)\ln\left(\frac{1}{\epsilon}\right) + k\ln\left(\frac{k}{\delta}\right)\right)\right).$$

When $k\ln(k) = O(d)$, the sample complexity is $\tilde{O}(\log(k)\frac{d}{\epsilon})$.

Personalized Learning yields an exponential improvement in the sample complexity with respect to the baseline; it improves the baseline's linear k dependence to logarithmic dependence, a drastic improvement for settings with a large number of players.

Subsequent works (Chen, Zhang, and Zhou 2018; Nguyen and Zakynthinou 2018) improve upon their suboptimal centralized learning algorithm using multiplicative weights approaches. In contrast to these works, we focus on the communication complexity of personalized and centralized learning. We build on the structure of these previously developed algorithms to obtain both sample and communication efficiency in our algorithms. Additionally, we consider communication-aware collaborative learning in the presence of classification noise. The previous work of (Qiao 2018) considers the collaborative PAC learning where some fraction of players behave truthfully while the remaining players behave adversarially. In addition to considering a different noise model than our work, (Qiao 2018) show that centralized learning is impossible in their setting and they do not consider communication complexity. To the best of our knowledge, no previous work has addressed the communication complexity of collaborative PAC learning.

Background

We now define notation and key concepts used in this paper. Let X denote the instance space and $Y = \{0, 1\}$ denote the set of possible labels. Let H denote a hypothesis class with finite VC-dimension d. We will assume the setting of realizable PAC learning, hence the target hypothesis h^* is in the hypothesis class H. The sample complexity of collaborative learning algorithms is defined in the standard way. The focus of this paper is on the communication cost of collaborative learning. We define the communication cost as the total number of samples transmitted between players in the execution of collaborative learning algorithms. To compute communication costs accurately and consistently, we carefully outline the implementation assumptions of our collaborative learning model. First, we define the completion of an algorithm as when each player is in possession of a classifier that has generalization error less than ϵ. Second, we assume that each player has computing power and a priori access to the hypothesis class H, ϵ, δ, and k. Third, we assume the broadcast model of communication, also known as the shared blackboard model, in which all players can observe all samples and bits transmitted to the center.

The second half of this paper considers collaborative learning in the presence of classification noise. In this setting, each player has their own distribution $D_i \sim X$ and their own classification noise rate $\eta_i < \frac{1}{2}$. Each player can generate instance-label pairs (x, y), where $x \sim D_i$, and with probability $1 - \eta_i$, $y = h^*(x)$, or with probability η_i, $y = \neg h^*(x)$. We let $\text{EX}_{\eta_i}(\cdot)$ denote the noisy distribution induced by a player's instance-label generating process. The center node orchestrating the learning process has full knowledge of players' noise rates but is not aware of the players' distributions.

The collaborative PAC learning criteria in the presence of noise is the same as in noiseless collaborative PAC learning except that the learned classifiers must generalize well on each individual player's *clean distribution*, that is, their distribution D_i without label noise. For $h \in H$, let $err_T(\text{EX}_{\eta_i}(\cdot), h)$ denote the empirical error of concept h on T points generated from $\text{EX}_{\eta_i}(\cdot)$. The definition of empirical error is standard and defined as

$$err_T(\text{EX}_{\eta_i}(\cdot), h) = \frac{1}{T}\sum_{j=1}^{T} \mathbf{1}_{\text{EX}_{\eta_i}(x_j) \neq h(x_j)}.$$

There are two types of generalization errors of h to consider. The first is the error of h on the *noisy distribution*, that is, the distribution D_i in the presence of label noise. The second is the error of h on the underlying clean distribution. In the classification noise setting, the learner only has access to samples from the noisy distribution, but the goal of learning is to generalize well with respect to the clean distribution. With access only to the noisy distribution, we use the generalization error with respect to the noisy distribution as a stepping stone in our analysis. The generalization error on the noisy data distribution, $\text{EX}_{\eta_i}(\cdot)$, is defined as

$$\text{err}_{D_i}(\text{EX}_{\eta_i}(\cdot), h) = \mathbb{E}_{T \sim D_i^T}[\text{err}_T(\text{EX}_{\eta_i}(\cdot), h)].$$

The generalization error on the clean data distribution, D_i,

is denoted $\mathrm{err}_{D_i}(h)$, and defined as follows,

$$\mathrm{err}_{D_i}(h) = \mathbb{E}_{T\sim D_i^T}[\mathrm{err}_T(h)] = \Pr_{x\sim D_i}[h(x) \neq h^*(x)].$$

In our algorithms and analysis, we use the classic empirical risk minimization (ERM) approach and sample complexity result of PAC learning with classification noise, in the single-player setting, recalled below.

Theorem 2 (Angluin and Laird 1987; Laird 1988)**.** *Let H denote a hypothesis class with finite VC-dimension d. Let D be a distribution on X and $\eta_i < \frac{1}{2}$. Let $EX_{\eta_i}(\cdot)$ denote an oracle that returns $(x, h^*(x))$ with probability $1-\eta_i$ or $(x, \neg h^*(x))$ with probability η_i. Given any sample S drawn from EX_η, an algorithm A that produces a hypothesis $h \in H$ that minimizes disagreements with S satisfies the PAC criterion, i.e. for any $\epsilon, \delta > 0$ and any distribution D on X, $\Pr_{S\sim D^m}[err_D(h) \geq \epsilon] \leq \delta$, with sample complexity*

$$m_{\epsilon,\delta,\eta_i} = O\left(\frac{d\log(1/\delta)}{\epsilon(1-2\eta_i)^2}\right).$$

We note that since the confidence parameter δ is handled in a standard fashion, for the duration of this paper we suppress δ dependency for clarity.

Communication-Aware Personalized Learning

We define the communication cost of a collaborative PAC learning algorithm as the total number of samples transmitted to the center. In contrast, the sample complexity reflects the total number of samples, whether transmitted or not, consumed by the algorithm. Our goal is to achieve communication efficiency, while retaining sample efficiency, in the personalized learning setting. In this section, we develop a personalized learning algorithm whose sample complexity matches that of Personalized Learning (Theorem 1) and whose communication cost is less than that of Personalized Learning, deeming our algorithm the best of both worlds.

Before describing our approach, we first compute the communication costs of the baseline approach and Personalized Learning. The personalized learning baseline approach is where each player draws $\tilde{O}(\frac{d}{\epsilon})$ examples locally from their own distributions and independently learns their own classifier. This baseline requires no communication to the center. Hence, simultaneous communication and sample efficiency is necessary for our algorithms to be meaningful as we are competing with a baseline whose communication complexity is zero. In other words, if both sample complexity and communication cost are a concern, it will only make sense to choose algorithms other than this baseline if the communication cost is not too much and the sample complexity is substantially lower.

The communication cost of Personalized Learning was not considered in previous works. We compute the communication complexity of Personalized Learning, in light of our implementation assumptions, in the following proposition.

Proposition 3. *The communication cost of Personalized Learning is*

$$\tilde{O}\left(\log(k)\frac{d}{\epsilon}\right)$$

samples plus $\tilde{O}\left(k\log\left(\frac{d}{\epsilon}\right)\right)$ additional bits of communication.

Proof. We describe the implementation details for Personalized Learning, described in Algorithm 1. Consider round j. In the first step, the center computes the number of samples to request from each player by drawing $m_{\epsilon/4,\delta'}/|N_j|$ samples from the uniform multinomial distribution. The center communicates this quantity to each player, costing $O(k\log(\frac{d}{\epsilon}))$ bits. The players then communicate their requested quantity of samples. By assumption of the broadcast model, each player can see the samples transmitted by other players so all players can learn a consistent hypothesis locally, costing no communication in this step. After learning the consistent hypothesis h_j, each player implements `TEST` locally, costing no communication. Afterwards, they communicate a single bit to the center indicating whether or not `TEST` passed with h_j, costing $O(k)$ bits of communication. Therefore, the total communication over $\log(k)$ rounds is $\tilde{O}(\log(k)\frac{d}{\epsilon})$ samples plus additional $O(k\log(k)\log(\frac{d}{\epsilon})) = \tilde{O}(k\log(\frac{d}{\epsilon}))$ bits of communication. □

Table 1 summarizes the sample and communication complexities of the baseline approach, Personalized Learning, and our algorithm, which we call Personalized Learning using Boosting. While our results state that there will be additional bits communicated to orchestrate these algorithms, they are not included in the tables as we are chiefly concerned with the number of samples communicated, as their representations can grow for large d. For completeness, we provide the full table, including additional bits communicated, in the Appendix.

The primary driver of communication inefficiency in Personalized Learning is the error parameter, ϵ. In applications such as the hospital scenario described in the introduction, learning highly accurate classifiers is crucial, hence ϵ is expected to be extremely small. Therefore, our goal is to improve communication complexity exponentially with respect to ϵ, while retaining the logarithmic k dependence granted by Personalized Learning. In particular, we show that our communication-efficient personalized learning algorithm has $O(\log(\frac{1}{\epsilon}))$ dependence in communication complexity.

Our approach to improving communication cost is to replace the first step in Personalized Learning with Distributed Boosting (Balcan et al. 2012), while keeping the remaining Personalized Learning algorithm intact. Distributed Boosting is a distributed implementation of AdaBoost ((Freund and Schapire 1997)) that learns a consistent hypothesis in $\tilde{O}(\log(\frac{1}{\epsilon}))$ rounds. We note that the objective of Distributed Boosting is to learn a classifier with error less than ϵ on the *mixture* of distributions. We recall the communication complexity of Distributed Boosting below.

Theorem 4 (Balcan et al. 2012)**.** *Any class H of finite VC-dimension d can be learned to error ϵ in $\tilde{O}(\log(\frac{1}{\epsilon}))$ rounds and $O(d)$ examples plus $O(k\log(d))$ bits of communication per round using the distributed boosting algorithm.*

	Sample Complexity	Samples Communicated
Baseline	$\tilde{O}(k\frac{d}{\epsilon})$	$\tilde{O}(1)$
Personalized Learning	$\tilde{O}(\log(k)\frac{d}{\epsilon})$	$\tilde{O}(\log(k)\frac{d}{\epsilon})$
Personalized Learning using Boosting	$\tilde{O}(\log(k)\frac{d}{\epsilon})$	$\tilde{O}(\log(k)d\log(\frac{1}{\epsilon}))$

Table 1: Sample and Communication Costs of Personalized Learning Variants

By using Distributed Boosting as the first step in Personalized Learning, by Theorem 4 we will achieve logarithmic dependence on ϵ in communication cost. However, we must be careful that we don't achieve this improved communication at the cost of higher sample complexity. To the best of our knowledge, the sample complexity of Distributed Boosting was not previously analyzed. We derive the sample complexity of Distributed Boosting in the next section, showing that Distributed Boosting can be implemented with the same sample complexity as AdaBoost.

Sample Complexity of Distributed Boosting

We first recall the sample complexity of AdaBoost (Freund and Schapire 1997). In AdaBoost, a large sample, denoted by S, is drawn from an unknown distribution. Throughout AdaBoost, S is perpetually resampled. The size of S, the size of the *reservoir* of points used in the AdaBoost routine, is the sample cost. To review the sample complexity of AdaBoost, we first recall the VC-dimension of the hypothesis class H after T rounds of boosting.

Lemma 5 (Freund and Schapire 1997)**.** *Suppose the weak learner in AdaBoost learns a classifier with constant error in each round. Then, $\tilde{O}(\ln(\frac{1}{\epsilon}))$ rounds of AdaBoost are needed to learn a classifier with zero training error.*

Let d_{boost} denote the VC-dimension of the hypothesis class after T rounds of boosting.

Lemma 6 (Freund and Schapire 1997)**.** *Let H denote the base class of hypotheses with VC-dimension d. After T rounds of boosting, the resulting hypothesis class has VC-dimension $d_{boost} = O(dT\log(T)) = \tilde{O}(dT)$.*

We recall the folklore result of the sample complexity of AdaBoost, which follows immediately from Lemma 5, Lemma 6, and realizable PAC sample complexity bounds.

Lemma 7. *The sample complexity of AdaBoost is*

$$m_{boost} = O\left(\frac{d_{boost}}{\epsilon}\right) = \tilde{O}\left(\frac{d}{\epsilon}\right).$$

We now show that the sample complexity of Distributed Boosting is the same as that of AdaBoost. In Distributed Boosting, there are k players implementing AdaBoost. Each player has a reservoir of points, S_i, from which the center resamples. The sample cost is the sum over the players' sample reservoirs, $\sum_{i=1}^{k} S_i$. From standard learning theory we know lower and upper bounds on $\sum_{i=1}^{k}|S_i|$, but the size with which to initialize each individual S_i so that the algorithm is correct was not previously analyzed. During each round of Distributed Boosting, the number of samples the center requests from a player can increase and in the original analysis of Distributed Boosting, S_i was defined to be ambiguously large (Balcan et al. 2012). We give clarity to the size of each S_i needed for Distributed Boosting. To do so, we first propose adding the following one-time preprocessing step to Distributed Boosting: let the center draw m_{boost} points from a uniform multinomial distribution to determine the sample size of each player's reservoir S_i. Communicating these sample sizes to the players cost $\tilde{O}(k\log(\frac{d}{\epsilon}))$ bits in total. This preprocessing step adds only a negligible cost to the bits communicated in Distributed Boosting. By initializing each reservoir in this way, we limit the total sample size to $\sum_{i=1}^{k}|S_i| = m_{\text{boost}}$.

Proposition 8. *The sample complexity of Distributed Boosting is*

$$O\left(\frac{d_{boost}}{\epsilon}\right) = \tilde{O}\left(\frac{d}{\epsilon}\right).$$

Proof. The derivation of the sample complexity of Distributed Boosting follows from the fact that Distributed Boosting is equivalent to AdaBoost with a single player and sample size $S = \cup_{i=1}^{k} S_i$. In this case, we know the sample complexity is $\tilde{O}(\frac{d}{\epsilon})$ by Lemma 7. By adding the preprocessing step described above, we restrict the sample complexity of the algorithm to m_{boost}. We now show that with m_{boost} samples, across players as prescribed by the preprocessing step, Distributed Boosting remains correct. The presampling step where the center draws from a multinomial distribution to determine the number of samples to request from each player remains unaffected by the new preprocessing step. However, in the sampling phase, it is possible that the center requests more points from a player than the player possesses. In this case, the player simply samples from their reservoir S_i i.i.d. proportional to the weights corresponding to the points. The weak learning step of Distributed Boosting is unaffected by the preprocessing step since it is still receiving a sample drawn i.i.d. from the boosting-weighted mixture of players. And finally, we note that the sample weights updating step remains unaffected. Therefore, $\tilde{O}(\frac{d}{\epsilon})$ samples suffice for Distributed Boosting and adding the preprocessing step to Distributed Boosting achieves the sample complexity. □

The result above reveals an important fact about the sample complexity of Distributed Boosting with k players – the sample complexity is surprisingly not dependent on k. Therefore, using Distributed Boosting, we can achieve sam-

ple and communication efficiency for the personalized learning setting, which we formalize in the next section.

Communication-Efficient Personalized Learning

Our approach to achieving a communication and sample efficient algorithm for the personalized learning setting is to replace the first step of Personalized Learning with the Distributed Boosting algorithm while leaving the remaining steps of Personalized Learning intact. We refer to our approach as Personalized Learning using Boosting. Using the sample complexity result on Distributed Boosting from the previous section, we compute the sample complexity of Personalized Learning using Boosting, showing that it is indeed equal (up to polylogarithmic terms) to the optimal sample complexity achieved by Personalized Learning.

Theorem 9. *The sample complexity of Personalized Learning using Boosting is*

$$\tilde{O}\left(\log(k)\frac{d}{\epsilon}\right)$$

when $k\ln(k) = O(d)$.

We now formally compute the communication complexity of Personalized Learning using Boosting, showing that it is an exponential improvement over the communication complexity of Personalized Learning with respect to ϵ.

Theorem 10. *The communication complexity of Personalized Learning using Boosting is*

$$\tilde{O}\left(\log(k)\left(d\log\left(\frac{1}{\epsilon}\right)\right)\right)$$

samples plus an additional $\tilde{O}(k\log(d)\log(\frac{1}{\epsilon}))$ *bits of communication.*

Proof. We consider a single round of our algorithm. The communication complexity of the first step is given in Theorem 4 as $\tilde{O}(d\log(\frac{1}{\epsilon}))$ examples plus $\tilde{O}(k\log(d)\log(\frac{1}{\epsilon}))$ bits of communication. Recall that each step in distributed boosting, all players learn the same weak learning classifiers locally. Therefore, when the distributed boosting algorithm completes, each player has all $\log(k)$ weak classifiers and can therefore sum them to create the final boosting classifier h_j, costing no communication. Using the boosting classifier in the TEST step, there is no communication needed as the players simply need to test the boosting classifier on T_j samples drawn from their own distributions. The players each send one bit of communication to the center indicating if they passed TEST or not, costing $O(k)$ bits for all k players. Therefore the total communication complexity over $\log(k)$ rounds is $\tilde{O}(\log(k)(d\log(\frac{1}{\epsilon}))$ samples plus $\tilde{O}(k\log(d)\log(\frac{1}{\epsilon}))+O(k) = \tilde{O}(k\log(d)\log(\frac{1}{\epsilon}))$ additional bits of communication. □

Communication-Aware Personalized Learning with Classification Noise

Thus far we have studied the communication cost of collaborative PAC learning without any assumptions of noise in the data. However, the presence of noise in data is often unavoidable in real-world learning scenarios. For instance, in the case where k hospitals work collaboratively to learn a diagnosis classifier, it is possible that a hospital's data has label noise from clerical errors or misdiagnoses. In this section, we consider communication-aware collaborative learning in the presence of classification noise, where each player has their own label noise rate $\eta_i < 1/2$ so that for any data point x drawn from their distribution D_i, with probability η_i they produce the wrong label and with probability $1-\eta_i$ they produce the correct label. We note that collaborative PAC learning in the presence of classification noise has not been previously analyzed. Thus, to build communication-efficient collaborative learning algorithms robust to classification noise, we first must analyze how to adapt collaborative learning to handle classification noise more generally. We note that the analysis and approaches in this section can be applied similarly to centralized learning in the presence of classification noise. Due to space constraints, we defer the details of centralized learning to the Appendix.

Personalized Learning with Classification Noise

Consider the baseline approach to personalized learning with classification noise, where the center requests $m_{\epsilon,\delta,\eta_i}$ samples from each player and learns an empirical risk minimizer (ERM), following exactly as in standard single-player PAC learning with classification noise (Theorem 2). In this case, the sample complexity is $\sum_{i=1}^{k} m_{\epsilon,\delta,\eta_i} = O(km_{\epsilon,\delta,\eta_{\text{MAX}}})$. The goal then is to develop a personalized learning algorithm with improved sample complexity.

We present our algorithm, Personalized Learning with Classification Noise (Algorithm 2), and show that it indeed improves upon the sample complexity of the baseline. The skeleton of our algorithm models that of (noiseless) Personalized Learning, but we make adjustments to handle classification noise. In the first and second steps of Personalized Learning, the center draws $m_{\epsilon/4,\delta'}$ samples and learns a consistent hypothesis. In contrast, in our algorithm, the center draws $m_{\epsilon/4,\delta',\bar{\eta}_{N_j}}$ points in total from the uniform mixture of players and learns an ERM hypothesis. When learning in the presence of classification noise, the existence of a hypothesis in the hypothesis class consistent with a sample generated from a noisy distribution is not guaranteed. Hence, our algorithm finds an ERM hypothesis instead of a consistent hypothesis. By Theorem 2 the ERM hypothesis has error $\epsilon/4$ when trained on $m_{\epsilon/4,\delta',\bar{\eta}_{N_j}}$ samples drawn from the noisy distribution. Finally, our CN−TEST subroutine differs from the TEST subroutine in Personalized Learning in that ours accounts for the individual noise rates of the players. Essentially, players must draw a factor of $\frac{1}{(1-2\eta_i)}$ more samples in CN−TEST than in TEST and the testing criterion is adjusted to reflect the relationship between drawing from noisy distribution and generalizing on the clean distribution.

The correctness of our algorithm will largely follow from the correctness results of Personalized Learning shown in (Blum et al. 2017), but with modifications to handle classification noise. We start by showing that even in the presence

Algorithm 2: Personalized Learning with Classification Noise

Input: H, k distributions $D_i \sim X$ with error rates $\eta_i < \frac{1}{2}$, $\delta' = \delta/2\log(k)$, $\epsilon > 0$
Output: $f_1, ..., f_k \in H$
Let $N_1 = \{1, ..., k\}$;
for $j = 1, ..., \lceil \log(k) \rceil$ **do**
  Draw sample S of size $m_{\epsilon/4,\delta',\bar{\eta}_{N_j}}$ from mixture $D_{N_j} = \frac{1}{|N_j|}\sum_{i \in N_j} D_i$;
  Select ERM hypothesis $h_j \in H$ on S;
  $G_j \leftarrow$ CN-TEST $(h_j, N_j, \epsilon, \delta')$;
  $N_{j+1} = N_j \setminus G_j$;
  for $i \in G_j$ **do**
    $f_i \leftarrow h_j$;
  end
end
return $f_1, ..., f_k$
Procedure CN-TEST (h, N, ϵ, δ)
  for $i \in N$ **do**
    Draw $T_i = O\left(\frac{\ln(\frac{|N|}{\delta})}{\epsilon(1-2\eta_i)}\right)$ samples from D_i;
  end
  return $\{i \mid err_{T_i}(EX_{\eta_i}, h_j) \leq \eta_i + \frac{3\epsilon}{4}(1-2\eta_i)\}$

of classification noise, the first and second steps in our algorithm yield a classifier that performs with error $\epsilon/4$ on the mixture.

Lemma 11. *The ERM h_j learned in Personalized Learning with Classification Noise has error no more than $\frac{\epsilon}{2}$ on at least half of the distributions in N_j.*

Next, we consider the CN-TEST subroutine, which tests if the learned ERM is a good classifier for each of the remaining players with respect to their underlying clean distribution. Recall that we only have access to their noisy data. Our analysis uses the following lemma from (Angluin and Laird 1987) that connects the generalization error of a concept h on the noisy distribution to the generalization error of h on the underlying clean distribution.

Lemma 12 (Angluin and Laird 1987)**.** *Let D be a distribution on X. Let η_i denote the classification noise rate and $h^* \in H$ denote the target function. Then,*

$$err_D(EX_{\eta_i}(\cdot), h) = \eta_i + err_D(h)(1 - 2\eta_i).$$

The following lemmas regarding the correctness of CN-TEST are due to multiplicative Chernoff bounds and Lemma 12.

Lemma 13. *With probability $1 - \delta'$, if h_j passes CN-TEST then $err_{D_i}(EX_{\eta_i}(\cdot), h_j) \leq \eta_i + (1 - 2\eta_i)\epsilon$. Hence, $err_{D_i}(h_j) \leq \epsilon$.*

Lemma 14. *With probability $1-\delta'$, if $err_{D_i}(EX_{\eta_i}(\cdot), h_j) \leq \eta_i + (1 - 2\eta_i)\frac{\epsilon}{2}$, then h_j passes CN-TEST. Hence, if $err_{D_i}(h_j) \leq \frac{\epsilon}{2}$, then h_j passes CN-TEST.*

We combine the above lemmas to show correctness of our algorithm, Personalized Learning with Classification Noise.

Proposition 15. *Personalized Learning with Classification Noise satisfies the personalized collaborative PAC learning criteria.*

Now, we compute the sample complexity of Personalized Learning with Classification Noise.

Proposition 16. *The sample complexity of Personalized Learning with Classification Noise is*

$$O\left(\log(k)\left(\frac{k\ln(k\log(k))}{\epsilon(1-2\eta_{MAX})} + \frac{d\ln(\log(k))}{\epsilon(1-2\eta_{MAX})^2}\right)\right).$$

When $k\ln(k) = O(d)$, the sample complexity simplifies to $\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{MAX})^2}\right)$.

Personalized Learning with Classification Noise improves upon the sample complexity of the baseline since it has logarithmic dependence on k instead of linear dependence on k. For settings with a large number of players, such as in a network of databases or a network of IoT devices, our algorithm can enjoy improved sample complexity. In fact, simplifying the sample complexity in Proposition 16 with respect to constant ϵ, δ, and η_{MAX}, and assuming $k\ln(k) = O(d)$, shows that our algorithm has $\tilde{O}(\log(k))$ overhead compared to the overhead of $\tilde{O}(k)$ from the baseline approach.

Communication-Efficient Personalized Learning with Classification Noise

We now return to the main goal of this section, which is to develop a communication-efficient personalized learning algorithm robust to classification noise. We first review the communication-efficient baseline approach, which is when each player draws $m_{\epsilon,\delta,\eta_i}$ samples from their own distribution and learns a classifier locally. The sample complexity of this baseline is $O\left(k\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$ and requires no samples nor bits of communication. To improve communication costs of personalized learning in the presence of noise, we build on Personalized Learning with Classification Noise developed in the previous section. We compute the communication cost of Personalized Learning with Classification Noise below.

Proposition 17. *The communication complexity of Personalized Learning with Classification Noise is*

$$\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{MAX})^2}\right)$$

samples and $\tilde{O}\left(k\log\left(\frac{d}{\epsilon(1-2\eta_{MAX})^2}\right)\right)$ additional bits of communication.

Proof. Recall that $\delta' = O(\delta/\log(k))$. In the first step, the center computes the number of samples to request from each player by drawing $m_{\epsilon/4,\delta'}/|N_j|$ samples from the uniform multinomial distribution. The center communicates this quantity to each player, costing $O(k\log(\frac{d}{\epsilon(1-2\bar{\eta}_{N_j})^2}))$ bits. The players then communicate their requested quantity of samples. Since we are in the broadcast model,

	Sample Complexity	Samples Communicated
Baseline	$\tilde{O}\left(k\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$	$\tilde{O}(1)$
Personalized Learning with CN	$\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$	$\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$
Personalized Learning with CN using Boosting	$\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$	$\tilde{O}\left(\log(k)d\log\left(\frac{1}{\epsilon(1-2\eta_{\text{MAX}})}\right)\right)$

Table 2: Sample and Communication Costs of Personalized Learning Variants with Classification Noise

each player observes all points communicated to the center, thereby allowing each player to learn an ERM hypothesis locally. Each player then implements CN−TEST locally, costing no communication. After completing CN−TEST, each player must send one bit to the center indicating their pass/fail result of CN−TEST, costing $O(k)$ bits in total. Over all $O(\log(k))$ rounds, the communication complexity is $O\left(\log(k)\frac{d\log(\log(k))}{\epsilon(1-2\bar{\eta}_K)^2}\right) = \tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)$ with an additional $O\left(\log(k)k\log\left(\frac{d}{\epsilon(1-2\bar{\eta}_{N_j})^2}\right)\right) + O(k\log(k)) = \tilde{O}\left(k\log\left(\frac{d}{\epsilon(1-2\eta_{\text{MAX}})^2}\right)\right)$ bits of communication. □

Table 2 summarizes the sample and communication costs of the baseline approach, Personalized Learning with Classification Noise, and our communication-efficient algorithm, Personalized Learning with Classification Noise using Boosting.

As discussed previously, we focus on the learning scenario where players want to learn highly accurate classifiers. Thus our goal is to develop an algorithm that improves dependence on $\frac{1}{\epsilon(1-2\eta_{\text{MAX}})}$ in samples communicated.

Our algorithm, Personalized Learning with Classification Noise using Boosting, is described as follows. We simply replace the first step of our noise-robust personalized learning algorithm, Personalized Learning with Classification Noise, with Distributed Agnostic Boosting (Chen, Balcan, and Chau 2016), while leaving the rest of Personalized Learning with Classification Noise intact. It is well known that boosting in the presence of classification noise is not straightforward. In fact, it has been shown that boosting the generalization error rate past the noise rate, so that $\eta_{\text{MAX}} > \epsilon$, is hard (Kalai and Servedio 2003). To avoid these issues, we restrict our attention to boosting the error ϵ up to the noise rate η_{MAX}, so that $\eta_{\text{MAX}} \leq \epsilon$. In this restricted regime, we use Distributed Agnostic Boosting from (Chen, Balcan, and Chau 2016), since classification noise is a special case of agnostic learning. Distributed Agnostic Boosting assumes access to a β-weak agnostic learner, which returns a hypothesis h so that $err_D(h) \leq \min_{h' \in H} err(h') + \beta$ (Chen, Balcan, and Chau 2016). We recall the sample and communication complexities of Distributed Agnostic Boosting below.

Theorem 18 (Chen, Balcan, and Chau 2016)**.** *Suppose Distributed Agnostic Boosting has access to a β-weak agnostic learner. Then, the sample complexity is*

$$\tilde{O}\left(\frac{d}{\epsilon^2(1/2-\beta)^2}\right).$$

It is well known that learning in the presence of classification noise is a special case of agnostic learning. We therefore derive the following corollary to the sample complexity of Distributed Agnostic Boosting in the restricted setting of classification noise.

Corollary 19. *Suppose Distributed Agnostic Boosting has access to a β-weak agnostic learner. Let β be a fixed constant. The sample complexity of Distributed Agnostic Boosting in the restricted setting of classification noise, where $\eta_{MAX} \leq \epsilon$, is*

$$\tilde{O}\left(\frac{d}{\epsilon(1-2\eta_{MAX})^2}\right).$$

We now recall the communication complexity of Distributed Agnostic Boosting.

Theorem 20 (Chen, Balcan, and Chau 2016)**.** *Suppose Distributed Agnostic Boosting has access to a β-weak agnostic learner. Then, Distributed Agnostic Boosting achieves error $\frac{2err_D(H)}{1/2-\beta} + \epsilon$ by using at most $O\left(\frac{\log(\frac{1}{\epsilon})}{(1/2-\beta)^2}\right)$ rounds, each communicating $O\left(\frac{d}{\beta}\log\left(\frac{1}{\beta}\right)\right)$ samples and $\tilde{O}\left(kd\log^2\left(\frac{d}{(1/2-\beta)\epsilon}\right)\right)$ words of communication.*

Similarly, we derive a corollary that holds specifically for the classification noise setting.

Corollary 21. *Suppose Distributed Agnostic Boosting has access to a β-weak agnostic learner. Let β be a fixed constant. The communication complexity of Distributed Agnostic Boosting in the restricted setting of classification noise, where $\eta_{MAX} \leq \epsilon$, consists of $O\left(\log\left(\frac{1}{\epsilon(1-2\eta_{MAX})}\right)\right)$ rounds, each communicating $O(d)$ samples and $\tilde{O}\left(kd\log^3\left(\frac{1}{\epsilon(1-2\eta_{MAX})}\right)\right)$ bits of communication.*

We derive the following sample and communication complexities of our algorithm, Personalized Learning with Classification Noise using Boosting.

Theorem 22. *The sample complexity of Personalized Learning with Classification Noise using Boosting is*

$$\tilde{O}\left(\log(k)\frac{d}{\epsilon(1-2\eta_{MAX})^2}\right).$$

Theorem 23. *The communication complexity of Personalized Learning with Classification Noise using Boosting is*

$$\tilde{O}\left(\log(k)d\log\left(\frac{1}{\epsilon(1-2\eta_{MAX})}\right)\right)$$

plus $\tilde{O}\left(kd\log^4\left(\frac{1}{\epsilon(1-2\eta_{MAX})}\right)\right)$ bits of communication.

Acknowledgements

This work was supported in part by the National Science Foundation under grants CCF-1815011, CCF-1934915, and CCF-1848966. This work was done while Shelby Heinecke was a student at UIC.

References

Angluin, D.; and Laird, P. D. 1987. Learning From Noisy Examples. *Machine Learning* 2(4): 343–370.

Balcan, M. F.; Blum, A.; Fine, S.; and Mansour, Y. 2012. Distributed Learning, Communication Complexity and Privacy. In Mannor, S.; Srebro, N.; and Williamson, R. C., eds., *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, 26.1–26.22. Edinburgh, Scotland: PMLR.

Blum, A.; Haghtalab, N.; Procaccia, A. D.; and Qiao, M. 2017. Collaborative PAC Learning. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 2392–2401. Curran Associates, Inc.

Chen, J.; Zhang, Q.; and Zhou, Y. 2018. Tight Bounds for Collaborative PAC Learning via Multiplicative Weights. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 3598–3607. Curran Associates, Inc.

Chen, S.-T.; Balcan, M.-F.; and Chau, D. H. 2016. Communication Efficient Distributed Agnostic Boosting. In Gretton, A.; and Robert, C. C., eds., *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, 1299–1307. Cadiz, Spain: PMLR.

Freund, Y.; and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55(1): 119 – 139. ISSN 0022-0000.

Kalai, A.; and Servedio, R. A. 2003. Boosting in the Presence of Noise. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC ?03, 195?205. New York, NY, USA: Association for Computing Machinery. ISBN 1581136749.

Laird, P. D. 1988. *Learning from Good and Bad Data*. Norwell, MA, USA: Kluwer Academic Publishers. ISBN 0-89838-263-7.

Nguyen, H.; and Zakynthinou, L. 2018. Improved Algorithms for Collaborative PAC Learning. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 7631–7639. Curran Associates, Inc.

Qiao, M. 2018. Do Outliers Ruin Collaboration? In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4180–4187. Stockholmsmässan, Stockholm Sweden: PMLR.