Asynchronous Differentially-Private Learning on Distributed Private Data

Abstract

Training data for machine learning can be located on multiple private geographically-scattered servers with different privacy settings. In this case, the number and the accessibility of the datasets might make it impossible to communicate with all private data owners simultaneously when training ML models. We develop asynchronous differentially-private algorithms for collaboratively training machine-learning models on multiple private datasets. The asynchronous nature of the algorithms implies that a central learner interacts with the private data owners one-on-one whenever they are available for communication without needing to aggregate query responses to construct gradients of the entire fitness function. Therefore, the algorithm efficiently scales to many data owners. The difference between the fitness for privacy-preserving machine-learning model and the fitness for trained machine-learning model in the absence of privacy concerns captures the cost of privacy. We prove that, by following the asynchronous privacy-preserving algorithms in this paper, the cost of privacy has an upper bound that is inversely proportional to the combined size of the training datasets squared and the sum of the privacy budgets squared. We validate the theoretical results with experiments on financial and health datasets.

Introduction

Unprecedented abundance of data has ignited a machine learning (ML) race that can potentially boost productivity and spur economic growth globally. However, the data required for responding to society's big challenges is often divided across multiple independent competing entities, e.g., financial or energy data is often scattered across servers for several service providers with competing interests. Regulatory frameworks, such as the GDPR, are increasingly restricting migration of private data across companies or even geographical boundaries for possible merger and training. This might restrict ML techniques from accessing the data in its entirety for training models. This motivates the development of distributed ML techniques with privacy guarantees.

Training data for machine learning can be located on multiple private geographically-scattered servers with different privacy settings. For instance, the training data can be gathered by Internet of Things (IoT) devices or hosted locally on smart devices with privacy settings enforced by users. Another example is cross-sector or -services machine learning using cross-nation datasets. In these cases, the sheer number of the datasets, legal boundaries, and accessibility of those datasets might make it impossible to communicate with all private data owners simultaneously when training ML models. Therefore, there is a need for asynchronous communication between the data owners and the learner (i.e., a central agent responsible for training ML models). Asynchronous communication implies that the learner can communicate with the data owners on a one-on-one basis without needing to wait for all data owners to respond. When using a gradient descent algorithm for training the ML model, the asynchronous communication raises an important challenge: the learner no longer knows the direction for the best model update based on all the training dataset; it can only infer the best update direction for the communicating data owner.

In this paper, we develop an asynchronous ML training algorithm. The learner updates the ML model based on only differentially-private (DP) gradient of the part of the fitness function that depends on the data possessed by the communicating data owner. To address the challenge of not knowing the direction for the best model update, the learner updates the ML model with small, yet constant, learning rates. The learner also shows inertia in updating its ML model so that it does not change the model significantly because of the gradient of just one data owner. These choices are motivated by that the learner is not overly confident that an update that is good for one data owner is also good for the others. The constant learning rate and the inertia of the learner allow the gradients of all the data owners to get mixed with each other across time so that the learner follow the direction for the best model update.

Note that, in this paper, we only investigate honest-but-curious threats in which the data owners do not trust the learner or each other for sharing private datasets while they trust that the learner trains the model correctly based on a pre-specified algorithm. For instance, in a financial setting, the central bank or government can be trusted for training the ML models but banks prefer not to share their data with

each other. In medical sciences or transportation, the government can also play the role of the central learner. For more general settings, incentives must be provided to ensure that the learner follows the training algorithm (Parkes and Shneidman 2004; Tanaka, Farokhi, and Langbort 2015).

The difference between the fitness function evaluated for privacy-preserving ML model and the fitness function evaluated for trained ML model without privacy concerns, or the degradation caused in the performance of ML models by the presence of DP noises, captures the cost of privacy. We prove that, by following the proposed asynchronous ML training algorithm in this paper, the cost of privacy is inversely proportional to the combined size of the training datasets squared and the privacy budgets squared. We validate the theoretical results on experiments on financial data. We use linear regression on a dataset of loan information from the Lending Club, a peer-to-peer lending platform, for setting setting interest rates of loans based on attributes, such as loan size and credit rating. We will also use regression models on a dataset of hospital visits by patients in the United States for determining the length of stay based on parameters, such as age, gender, and diagnosis. We show that, for collaboration among large numbers of private data owners, i.e., more than 10 data owners with at least 10,000 records, and with relatively large privacy budgets, i.e., privacy budgets greater than 1, the performance of the private ML model can beat the performance of a model that is trained with no collaboration. Therefore, we establish the value of collaboration in ML between multiple private data owners.

In summary, this paper contains the following contributions:

- Developing an asynchronous algorithm for training ML models with DP responses to gradient queries based on distributed private datasets; see Algorithm 1 and Theorem 1 in Section .
- Proving that the cost of privacy, the difference of the fitness for privacy-preserving ML model and the fitness for trained ML model in the absence of privacy concerns, is inversely proportional to the combined size of the training datasets squared and the privacy budgets squared; see Theorem 2 in Section .
- Validating the theoretical results by evaluating the asynchronous privacy-preserving ML algorithms on financial data for setting interest rates of loans and on health data for determining the length of hospital stay for patients; see Section .

Related Work

Secure Multi-Party Computation and Homomorphic Encryption We may use secure multi-party computation, for instance based on homomorphic encryption, to eliminate privacy concerns when training ML models on multiple private datasets (Lindell and Pinkas 2000; Du, Han, and Chen 2004; Vaidya and Clifton 2002; Vaidya, Kantarcıoğlu, and Clifton 2008; Jagannathan and Wright 2005). The secure methods can be utilized for both training (Bonawitz et al. 2017; Graepel, Lauter, and Naehrig 2012; Hunt et al. 2018; Li et al. 2017; Aono et al. 2018) and evaluation (Gilad-Bachrach et al. 2016). Although powerful in theory, secure multi-party computation and homomorphic encryption introduce massive computation and communication overheads resulting in inefficiency of such training algorithms. They also do not fully eliminate the risk of privacy breaches, e.g., risks associated with membership inference and model inversion attacks still remain if these algorithms are not pared with other privacy-preserving techniques, such as DP.

ML with Differential Privacy ML with DP has been used in the past alleviate privacy risks (Sarwate and Chaudhuri 2013; Zhang et al. 2012; Chaudhuri and Monteleoni 2009; Zhang, Rubinstein, and Dimitrakakis 2016). These approaches however require merging the private datasets for training. They rely on obfuscating the ML model using DP noise once the training on the aggregated data is performed. Alternatively, they train the ML model based on the obfuscated, yet merged data. These studies do not investigate merging the data in a privacy-preserving manner or providing a certain level of autonomy to the data owners by only requiring responses to some queries on the private dataset.

Distributed and/or Collaborative Privacy-Preserving ML Distributed privacy-preserving ML proposes the use of DP gradients for training ML models (Zhang and Zhu 2017; Huang et al. 2018; Abadi et al. 2016; Zhang, He, and Lee 2018; Wu et al. 2020; McMahan et al. 2017; Shokri and Shmatikov 2015). Noisy DP gradients can be used to train ML models with convex and non-convex fitness functions (Wu et al. 2020; Shokri and Shmatikov 2015; McMahan et al. 2017). An important aspect of these studies is that they sometimes use better DP composition methods, such as moment accountant, for reducing the scale of the DP noise (Abadi et al. 2016). These studies however propose synchronous updates in which the ML model must be updated according to the contributions of all the data owners simultaneously (rather than a subset of them). This assumption can prohibit the use of the above distributed or collaborative ML training algorithms in the presence of numerous data owners. We particularly extend the setup of (Wu et al. 2020) by allowing the learner to communicate with the datasets in a one-on-one basis whenever they are available. The availability for communication is particularly modelled using Poisson point processes. These processes are often utilized for analysis of asynchronous multi-agent systems and are shown to mimic practical scenarios (Ram, Nedić, and Veeravalli 2009; Farokhi and Johansson 2014; Heidelberger and Trivedi 1982; Lagunoff and Matsui 1997).

Asynchronous Distributed Optimization and Machine Learning Distributed asynchronous optimization algorithms can be used for training ML models (Srivastava and Nedic 2011; Touri, Nedič, and Ram 2010; Aybat, Wang, and Iyengar 2015; Ram, Nedić, and Veeravalli 2009; Bedi and Rajawat 2018; Hong and Chang 2017). This is because we can rewrite distributed ML training as a distributed optimization problem with private datasets represented as various parts of the fitness function. These algorithms are however generic and do not address the issue

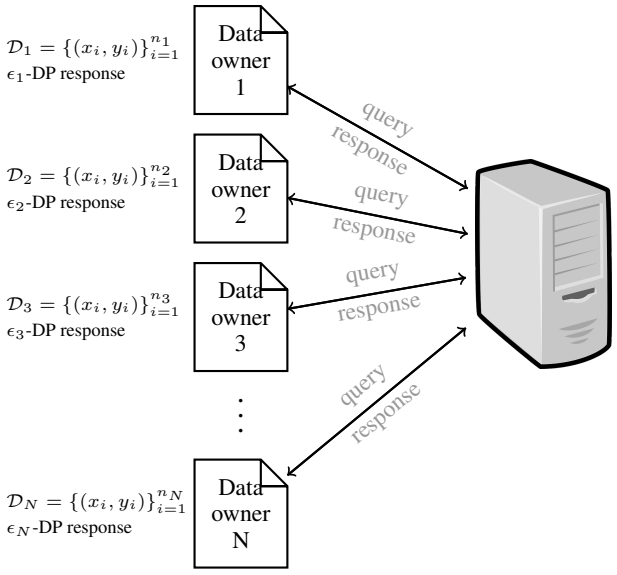

Figure 1: Communication structure between a central learner and multiple data owners with private datasets.

of selecting learning rate for ML training with DP gradients and forecasting the quality of the trained ML model based on dataset sizes and privacy budgets. Forecasting the performance of privacy-preserving ML algorithms can be used to understand the value of collaboration between distributed private datasets. Without such forecasts the private data owners might need to forgo their private datasets so that a trusted third-party can compare the performance of the private ML model with the ML model trained in absence of privacy concerns (as otherwise there is no ground truth for comparison in general). Asynchronous optimization has been also utilized in the past for ML purposes; see, e.g., (Mnih et al. 2016; Smyth, Welling, and Asuncion 2009; McMahan and Streeter 2014). These studies however do not consider additive DP noises and their impact on the quality of trained ML models.

Asynchronous ML Training with DP

We consider $N \in \mathbb{N}$ private data owners connected to a central learning node, referred to as learner, responsible for training a ML model. Figure 1 depicts the communication structure between the learner and the private data owners. The set of the data owners is denoted by $\mathcal{N} := \{1, \dots, N\}$. The data owners possess a private training dataset composed of inputs x_i and outputs y_i. The dataset is denoted by $\mathcal{D}_i := \{(x_i, y_i)\}_{i=1}^{n_i} \subseteq \mathbb{X} \times \mathbb{Y} \subseteq \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}$.

An ML model is a meaningful relationship between inputs and outputs in a training dataset. The ML model is $\mathfrak{M}(\cdot;\theta)$ for some mapping $\mathfrak{M} : \mathbb{X} \times \mathbb{R}^{p_\theta} \to \mathbb{Y}$ with $\theta \in \mathbb{R}^{p_\theta}$ denoting the parameters of the ML model. The learner in Figure 1 aims to train the ML model $\mathfrak{M}(\cdot;\theta)$ based on the available training datasets $\mathcal{D}_i$, $\forall i \in \mathcal{N}$, by solving the optimization problem in

$$\theta^* \in \arg\min_{\theta \in \Theta} f(\theta), \tag{1}$$

where $\Theta := \{\theta \in \mathbb{R}^{p_\theta} \mid \|\theta\|_\infty \leq \theta_{\max}\}$ and $f : \mathbb{R}^{p_\theta} \to \mathbb{R}$ is the fitness for ML model parameter θ, i.e., the fitness of ML model $\mathfrak{M}(\cdot;\theta)$ for relating the inputs and outputs in the training dataset $\cup_{j \in \mathcal{N}} \mathcal{D}_j$, given by

$$\begin{aligned} f(\theta) :=& g(\theta) + \frac{1}{n} \sum_{i \in \mathcal{N}} \sum_{\{x,y\} \in \mathcal{D}_i} \ell(\mathfrak{M}(x;\theta), y) \\ =& g(\theta) + \frac{1}{n} \sum_{\{x,y\} \in \bigcup_{j \in \mathcal{N}} \mathcal{D}_j} \ell(\mathfrak{M}(x;\theta), y). \end{aligned} \tag{2}$$

In the fitness (2), $g(\theta)$ is a regularizing term, $\ell(\mathfrak{M}(x;\theta), y)$ is a loss function capturing the distance between the output of the ML model $\mathfrak{M}(x;\theta)$ and the true output y, and $n = \sum_{j \in \mathcal{N}} n_j$. Finally, note that we can select a large enough $\theta_{\max}$ so that, if desired, training on Θ does not add any conservatism (in comparison to the unconstrained case). We make the following standing assumptions throughout the paper.

Assumption 1. *The regularizing term $g(\theta)$ is σ strongly convex in θ and the loss function $\ell(\mathfrak{M}(x;\theta), y)$ is convex in θ.*

Assumption 2. *The following properties hold:*

1. $\Xi_g := \sup_{\theta \in \Theta} \|\nabla_\theta g(\theta)\|_2 < \infty$;
2. $\Xi := \sup_{\theta \in \Theta} \sup_{(x,y) \in \mathbb{X} \times \mathbb{Y}} \|\nabla_\theta \ell(\mathfrak{M}(x;\theta), y)\|_2 < \infty$.

We consider the case where the data owners cannot share their private data. In this case, the learner must submit queries $\mathfrak{Q}_i(\mathcal{D}_i; k) \in \mathcal{Q}$ to data owners $i \in \mathcal{N}$ iteratively in which k identifies the iteration number and $\mathcal{Q}$ is the output space of the queries. The data owner $i \in \mathcal{N}$ then provides DP response $\overline{\mathfrak{Q}}_i(\mathcal{D}_i; k) \in \mathcal{Q}$ to the query $\mathfrak{Q}_i(\mathcal{D}_i; k) \in \mathcal{Q}$.

Typically, numerous data owners might participate in private ML learning. In addition to ML training, the data owners might have their own internal responsibilities and thus might not be available in synchronized times to communicate with the learner. The learner might also not have the communication and computational capacities required to interact with all the data owners at the same time. Therefore, we consider an asynchronous algorithm for interaction between the data owners and the learner. We model the internal clock of the data owner by Poisson point processes with rates of one. At random times, the Poisson processes instigate communication between the data owners and the learner on a one-on-one basis. The Poisson process model is often utilized for analysis of asynchronous multi-agent systems (Ram, Nedić, and Veeravalli 2009; Heidelberger and Trivedi 1982; Lagunoff and Matsui 1997).

Let the time instants in which the data owners communicate with the learner be given by

$$0 = t_1 \leq t_2 \leq \cdots \leq t_k \leq \cdots \leq t_T.$$

At each time instant t_k, $k \in \mathbb{N}$, one the data owners at random communicates with the learner. We use the notation

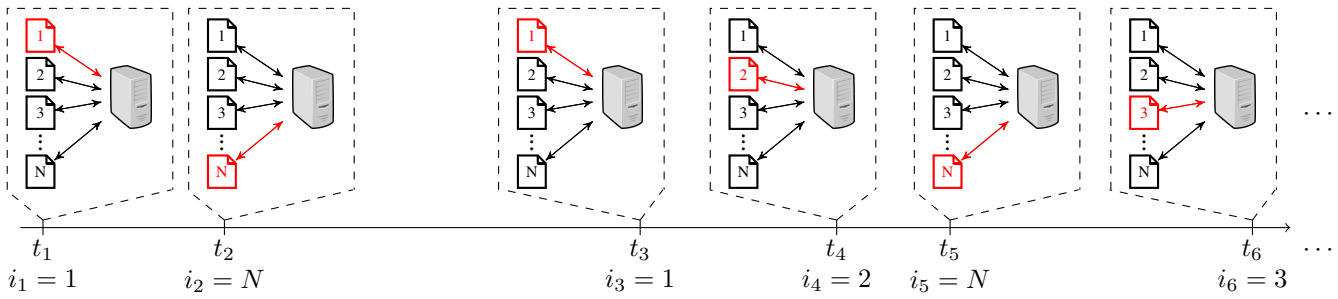

Figure 2: A random asynchronous sequence of ML model updates. In each step, the learner submit a gradient query to one of data owners i_k, marked in red. The data owner provides a DP response that is used to train the ML model.

$i_k \in \mathbb{N}$ to denote the index of that data owner. This mechanism is illustrated in Figure 2. We can interpret this mechanism in multiple ways:

- **Broadcasting by the Learner**: In this scenario, we can assume that the learner, in regular time intervals, broadcasts gradient queries to all data owners (some might be listening while others not). Whenever one of the data owners responds, the index k is incremented. Let t_k denote the time at which the communication takes place and i_k denote the index of the communicating data owner. Since there is only one communication channel available, if two or more data owners simultaneously respond a collision occurs and the responses are lost. Therefore, at any given time t_k, only one data owner can communicate with the learner. The data owners can measure the channel's availability or use timed communication protocols to avoid collision.
- **Request for Update by the Data Owner**: In this scenario, we can assume that the leaner is constantly listening for requests of update. Whenever a data owner submits such a request, the index k is incremented with t_k denoting the time and i_k denoting the index of the data owner. At this point, the learner only communicates that data owner until the update is over.

Assumption 3. *$T \in \mathbb{N}$ is the maximum number of iterations for communication between data owners and learner.*

Definition 1 (Differential Privacy)**.** *Responses of data owner $\ell \in \mathcal{N}$ are ϵ_ℓ-differentially private (or ϵ_ℓ-DP) over the horizon T if*

$$\mathbb{P}\Big\{(\overline{\Omega}_\ell(\mathcal{D}_\ell;k))_{k:i_k=\ell} \in \mathcal{Y}\Big\} \leq \exp(\epsilon_\ell)\mathbb{P}\Big\{(\overline{\Omega}_\ell(\mathcal{D}'_\ell;k))_{k:i_k=\ell} \in \mathcal{Y}\Big\},$$

where $\mathcal{Y}$ is any Borel-measurable subset of $\mathcal{Q}^{|\{k:i_k=\ell\}|}$, and $\mathcal{D}_\ell$ and $\mathcal{D}'_\ell$ are two adjacent datasets differing at most in one entry, i.e., $|\mathcal{D}_\ell \setminus \mathcal{D}'_\ell| = |\mathcal{D}'_\ell \setminus \mathcal{D}_\ell| \leq 1$.

The learner processes all the received DP response to generate a privately-trained ML model. In this paper, we use Algorithm 1 for generating queries and using the DP responses for computing the ML model. In Algorithm 1, the learner updates one central ML model, i.e., $\theta_{L,k}$, and N copies of it for each data owners, i.e., $\theta_{i,k}$ for each $i = 1, \ldots, N$. A copy is only updated when the corresponding data owner is communicating with the learner. This is to keep track of the updates for each data owner. In line 7 of Algorithm 1, the learner updates the ML model with small, yet constant, learning rates. The learner also shows inertia in updating its ML model so that it does not change the model significantly because of the gradient of just one data owner; see the update for $\theta_{L,k}$. These steps are motivated that the learner is not overly confident that an update that is good for one data owner is also good for the others because the learner can only access the gradient of the part of the fitness function that depends on the data possessed by the communicating data owner. The constant learning rate and the inertia of the learner allow the gradients of all the data owners to get mixed with each other across time so that the learner follow the direction for the best model update.

Theorem 1. *The policy of data owners in line 6 of Algorithm 1 for responding to the queries over the horizon $\{1, \ldots, T\}$ is ϵ_i-DP, $\forall i \in \mathcal{N}$, if $w_i(k)$ are statistically independent Laplace noises with scale $2\Xi T/(n_i\epsilon_i)$.*

Proof. See Appendix . □

Performance of Private ML Models

For Algorithm 1, we can prove the following convergence result under the assumptions of strong convexity and smoothness of the ML fitness function.

Theorem 2. *For any N, there exist constants*[1] *$c_1, c_2, c'_1, c'_2 > 0$ such that the iterates of Algorithm 1 satisfy*

$$\mathbb{E}\{\|\theta_{L,T} - \theta^*\|_2^2\} \leq c_1\sqrt{\frac{1}{T^2} + N\sum_{i\in\mathcal{N}}\left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i}\right)^2} + c_2\left(\frac{1}{T^2} + N\sum_{i\in\mathcal{N}}\left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i}\right)^2\right). \quad (3)$$

[1] See the proof of the theorem for the exact expression of the constants.

Algorithm 1 Asynchronous ML training algorithm with distributed private datasets using DP gradients for strongly-convex smooth fitness cost. The pink areas illustrate the responsibility of the learner and the cyan area captures the responsibility of the data owner.

Require: $T \in \mathbb{N}$, $\rho \in \mathbb{R}_{\geq 0}$

Ensure: $(\theta_{1,k}, \theta_{2,k}, \ldots, \theta_{N,k}, \theta_{L,k})_{k=1}^{T}$

1: Initialize $\theta_{1,0} = \theta_{2,0} = \cdots = \theta_{N,0} = \theta_{L,0} = 0$

2: **for** $k = 1, \ldots, T$ **do**

3: Randomly at uniform select data owner i_k

4: Compute $\bar{\theta}_k \leftarrow (\theta_{L,k-1} + \theta_{i_k,k-1})/2$

5: Submit gradient query to data owner i:

$$\mathcal{Q}_{i_k}(\mathcal{D}_{i_k}; \bar{\theta}_k) := \frac{1}{n_{i_k}} \sum_{\{x,y\} \in \mathcal{D}_{i_k}} \nabla_\theta \ell(\mathfrak{M}(x;\theta), y)$$

6: Provide DP response

$$\overline{\mathcal{Q}}_{i_k}(\mathcal{D}_{i_k}; \bar{\theta}_k) = \mathcal{Q}_{i_k}(\mathcal{D}_{i_k}; \bar{\theta}_k) + w_{i_k}(k)$$

7: Compute

$$\theta_{i_k,k} = \Pi_\Theta \left[\bar{\theta}_k - \frac{N\rho}{T^2\sigma} \left(\frac{1}{2N} \nabla_\theta g(\bar{\theta}_k) + \frac{n_{i_k}}{n} \overline{\mathcal{Q}}_{i_k}(\mathcal{D}_{i_k}; \bar{\theta}_k) \right) \right]$$

$$\theta_{L,k} = \Pi_\Theta \left[\bar{\theta}_k - \frac{(N-1)\rho}{NT^2\sigma} \nabla_\theta g(\bar{\theta}_k) \right]$$

8: **end for**

and

$$\mathbb{E}\{f(\theta_{L,T})\} - f(\theta^*) \leq c_1' \sqrt{\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2} + c_2' \left(\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2 \right). \quad (4)$$

Proof. See Appendix . □

For large enough learning horizon T, the upper bound (3) takes the form of

$$\mathbb{E}\{\|\theta_{L,T} - \theta^*\|_2^2\} \leq \frac{\bar{c}_1}{n} \sqrt{\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2}} + \frac{\bar{c}_2}{n^2} \left(\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2} \right), \quad (5)$$

where $\bar{c}_1 = \sqrt{8N} c_1$ and $\bar{c}_2 = 8N c_2$. Similarly, for large T, the upper bound (3) takes the form of

$$\mathbb{E}\{f(\theta_{L,T})\} - f(\theta^*) \leq \frac{\bar{c}_1'}{n} \sqrt{\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2}} + \frac{\bar{c}_2'}{n^2} \left(\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2} \right), \quad (6)$$

where again $\bar{c}_1' = \sqrt{8N} c_1'$ and $\bar{c}_2' = 8N c_2'$. This takes the form of the performance bound in (Wu et al. 2020). Under the assumption that all the data owners have equal privacy budgets $\epsilon_i = \epsilon$, $\forall i$, the bound in (6) scales as ϵ^{-2}.

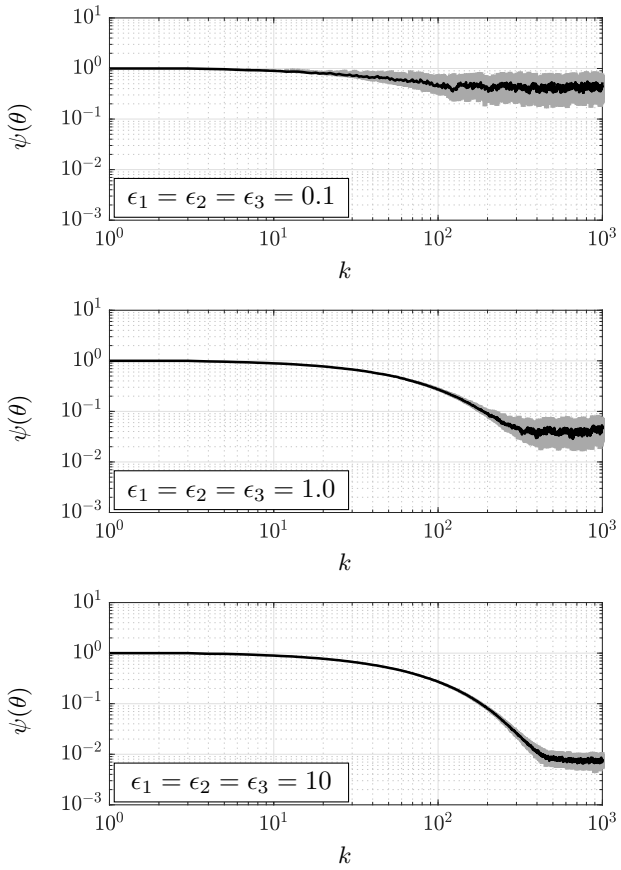

Figure 3: Percentile statistics of relative fitness of 100 runs of Algorithm 1 for learning lending-interest-rates versus the iteration number k for a learning horizon of $T = 1,000$ iterations with three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The gray area illustrates the range of 25% to 75% percentiles and the black line shows the median of relative fitness.

This bound matches the lower and upper bounds in (Bassily, Smith, and Thakurta 2014) for strongly convex loss functions. The same outcome also holds if $N = 1$ and $\epsilon_1 = \epsilon$ which captures centralized privacy-preserving learning.

We can introduce the cost of privacy (CoP) as the difference of the fitness for privacy-preserving ML model and the fitness for trained ML model in the absence of privacy concerns. The inequalities in (5) and (6) show that CoP is inversely proportional to the combined size of the training datasets squared and the sum of the privacy budgets squared.

Experimental Validation

In this section, we investigate the performance of Algorithm 1 on real datasets from finance and medical science. In our experiments, the datasets have significantly different sizes and the size of the training datasets influence the performance of both non-private and private ML models. Hence, we factor out the effects of the size of the training datasets on the performance of the learning by only considering the relative fitness, defined as $\psi(\theta) := f(\theta)/f(\theta^*) - 1$.

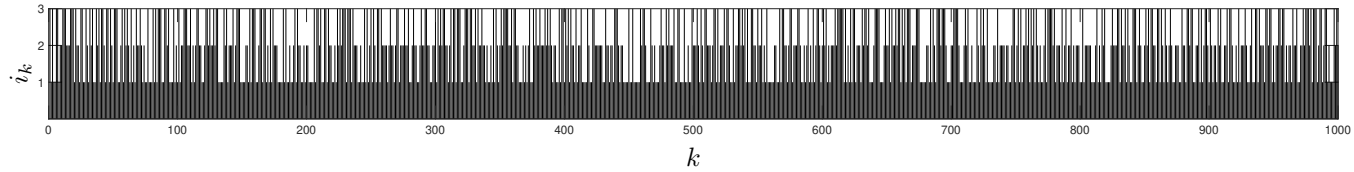

Figure 4: Example of communication timing for the asynchronous learning in Algorithm 1 for learning lending-interest-rates, illustrating i_k versus the iteration number k.

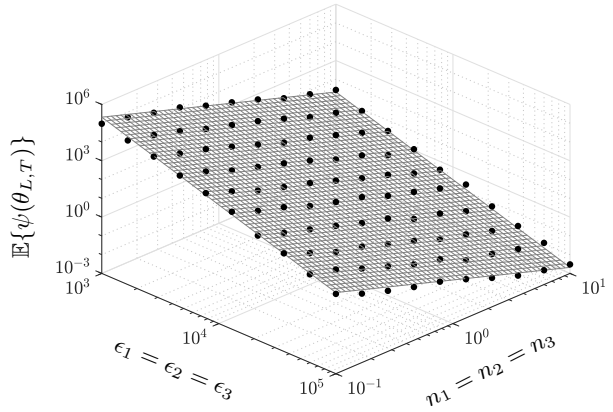

Figure 5: Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The mesh surface illustrates the bound in (6) with $\bar{c}_1' = 0$ and $\bar{c}_2' = 2.1 \times 10^9$.

This measure captures the quality of any ML model θ in comparison to the non-private ML model θ^* in terms of the fitness in (2). By definition, $\psi(\theta) \geq 0$ for any ML model θ. The larger $\psi(\theta)$, the worse performance of ML model θ is in comparison with the non-private ML model θ^*.

Lending Dataset (Financial)

We first train a linear regression model on lending datasets as an example of automating banking processes requiring access to sensitive private datasets.

Dataset Description and Pre-Processing We use a dataset of anonymized loan application information from roughly 890,000 individuals (Kaggle 2018). We remove unique identifiers, such as id and member id, and irrelevant attributes, such as the URL addresses. We endeavour to train a linear regression model on this dataset. The input to the regression model are loan information, such as loan size, and applicant information, such as credit rating, state of residence, and age. The model estimates the annual interest rate for the loans. We encode categorical attributes, such as state of residence and loan grade, with integer numbers.

In order to improve the numerical stability of the algorithm, we use Principal Component Analysis (PCA) to perform feature selection. We select the top ten important features. For this step, we only use the last ten-thousand entries of the dataset. We can assume that these entries are known to the learner and thus do not violate the distributed nature of the algorithm. This would have been a restrictive assumption if the learner used the entire dataset for the PCA (because the data owners must have agreed to perform PCA in collaboration without privacy concerns, which is contradiction with their original interest for privacy-preserving ML). Using the PCA, the learner can construct a dictionary for feature selection using the eigenvectors corresponding to the most important features and communicate it to private data owners.

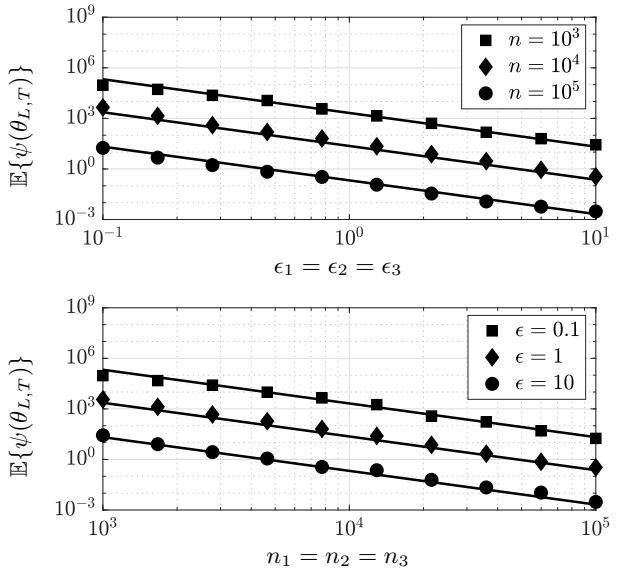

Figure 6: Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations versus the privacy budget [top] and the size of the datasets [bottom]. The solid line illustrates the bound in (6) with $\bar{c}_1' = 0$ and $\bar{c}_2' = 2.1 \times 10^9$.

Experiment Setup and Results We start with an experiment evaluating the outcome of collaborations between $N = 3$ banks. We use the linear regression model $y = \mathfrak{M}(x;\theta) := \theta^\top x$ with θ denoting the model parameters. The fitness function is given by $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 10^{-5}\theta^\top\theta$. The first data owner is assumed to possess the first n_1 entries of the dataset. The second data owner owns entries ranging from $n_1 + 1$ to $n_1 + n_2$. Finally, the third data owner has access to entries between $n_1 + n_2 + 1$ to $n_1 + n_2 + n_3$ as its private dataset.

We start with demonstrating the convergence of Algorithm 1 when $n_1 = n_2 = n_3 = 250,000$. Figure 3 illustrates the percentile statistics of the relative fitness $\psi(\theta_{L,k})$ for 100 runs of Algorithm 1 versus the iteration number k for the learning horizon $T = 1,000$. Note that, in Algorithm 1, only one of the data owners communicates with

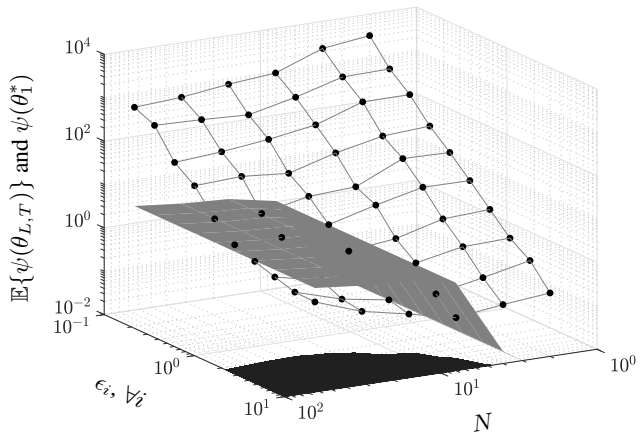

Figure 7: Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_i, \forall i$, and the number of collaborating data owners N. The solid gray surface shows the relative fitness of the non-private ML model θ_1^*, $\psi(\theta_1^*)$, constructed based on only the private data of the first data owner. If the relative fitness of Algorithm 1 is smaller than the relative fitness of the non-private ML model θ_1^*, collaboration benefits the first data owner (illustrated by the black region at the bottom of the figure).

the learner in each iteration. Figure 4 illustrates an example of communication timing for the asynchronous learning in Algorithm 1, illustrating i_k versus the iteration number k. Recalling the stochastic nature of the algorithm, due to the DP noise in query responses, the relative fitness varies for each run of the algorithm. The gray area in Figure 3 shows 25%–75% percentiles of the relative fitness. The black solid lines in Figure 3 show the median of relative fitness versus the iteration number. Evidently, the median decreases across time until the algorithm converges to a neighbourhood of the solution of (1). The relative fitness of the trained model also improves as $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases. Note that smaller privacy budgets also increase the variations in the relative fitness (i.e., larger gray area).

Figure 5 illustrates the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the size of the private datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The mesh surface shows the bound in (6) with $\bar{c}_1' = 0$ and $\bar{c}_2' = 2.1 \times 10^9$. This figure clearly shows the tightness of the result of Theorem 2. Note that, as expected, the relative fitness rapidly improves as either the sizes of the datasets $n_1 = n_2 = n_3$ or the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases.

Let us isolate the effects of the size of the datasets and the privacy budgets. Figure 6 shows the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ [top] and the size of the datasets $n_1 = n_2 = n_3$ [bottom]. In this figure, the markers (i.e., ■, ♦, and ●) are from the experiments and the solid show the bound in (6).

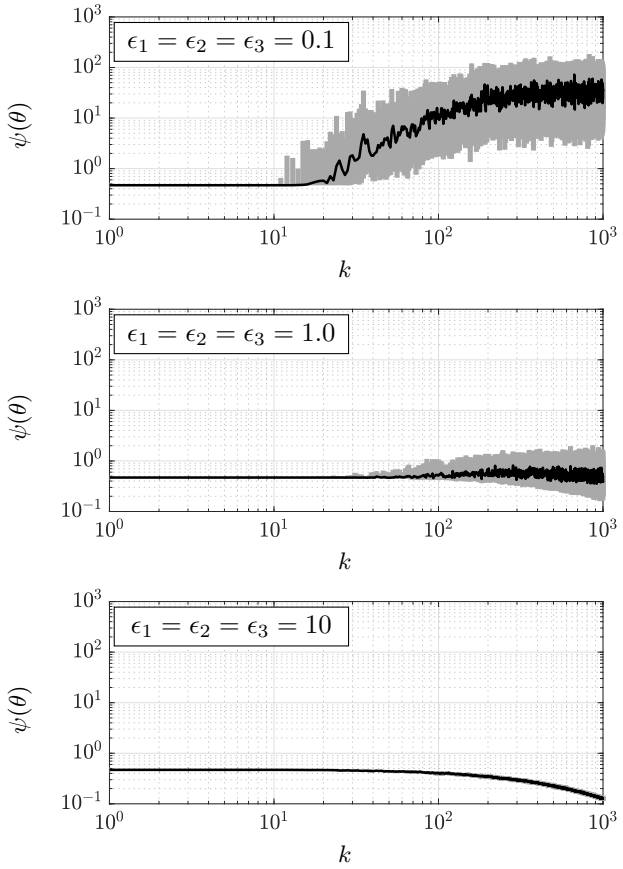

Figure 8: Percentile statistics of relative fitness of 100 runs of Algorithm 1 for learning length of stay at hospital versus the iteration number k for a learning horizon of $T = 1,000$ iterations with three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The gray area illustrates the range of 25% to 75% percentiles for the relative fitness and the black line shows the median of relative fitness.

For both these cases, the bounds in Theorem 2 are tight fits. Therefore, the theoretical results in Theorem 2 matches the experiments.

Finally, let us demonstrate the value of collaboration between among many banks. Consider an experiment with N banks each with $n_i = 10,000$ records collaborating to train a regression model.

Figure 7 shows the average relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_i, \forall i$, and the number of the collaborating data owners N. The solid gray surface shows the relative fitness of the non-private ML model θ_1^*, $\psi(\theta_1^*)$, constructed based on only the private data of the first data owner. Note that $\psi(\theta_1^*)$ is not random (as its construction does not require DP noise) and is not a function of ϵ_i. If the relative fitness of Algorithm 1 is smaller than the relative fitness of the non-private ML model θ_1^*, collaboration benefits the first data owner, which is illustrated by the black region at the bottom of the figure. Evidently, the first data owner benefits from collaboration if there are more than 5 data owners with privacy budgets greater than or equal to

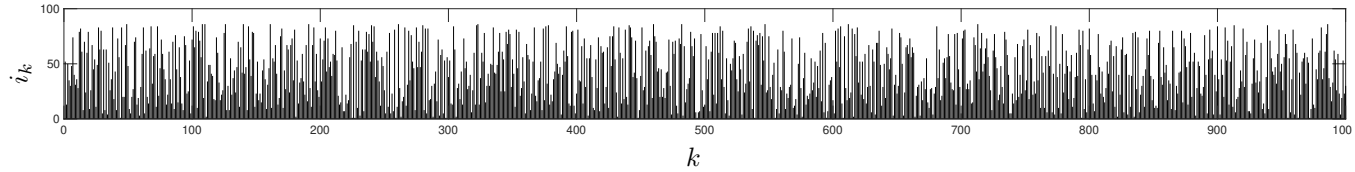

Figure 9: Example of communication timing for the asynchronous learning in Algorithm 1 for learning length of stay at hospital, illustrating i_k versus the iteration number k.

10 or if there are more than 100 data owners with privacy budgets greater than or equal to 2.5.

Medical Data

Now, we use the hospital admission and discharge dataset from the New York State to validate the theoretical results.

Dataset Description and Pre-Processing The dataset contains hospital visit and discharge information from nearly 2,350,000 de-identified patients including information, such as characteristics, diagnoses, treatments, services, and charges. This dataset is made public by the Bureau of Health Informatics at the New York State Department of Health (New York State Department of Health 2015). We train a linear regression model, as in the previous subsection, with inputs, such as age, gender, race, ethnicity, diagnosis code, procedure code, and drug code, to automatically determine the length of stay. This can be used as a tool for determining the capacity of hospitals in the future based on currently admitted patients. Similarly, we encode categorical attributes, such as gender and ethnicity, with integer numbers. We also remove attributes, such as total charges and costs, as well as irrelevant attributes, such as the postcode. Similar to the lending data, in order to improve the numerical stability of the algorithm, we perform the PCA to balance the features. We do so based on the last fifty-thousand entries of the dataset to ensure that the feature selection does not violate the distributed nature of the algorithm.

Experiment Setup and Results The data in (New York State Department of Health 2015) is tagged by the hospital name and code. There are 213 hospitals in the dataset. We focus on 86 hospital with at least 10,000 records.

We demonstrate the performance of the iterates of Algorithm 1. Figure 8 illustrates the percentile statistics of the relative fitness $\psi(\theta_{L,k})$ for 100 runs of Algorithm 1 versus the iteration number k for the learning horizon $T = 1,000$. Figure 9 illustrates an example of communication timing for the asynchronous learning in Algorithm 1, illustrating i_k versus the iteration number k. At each iteration, only one of the 86 data owners communicates with the learner. The gray area in Figure 8 shows 25%–75% percentiles of the relative fitness and the black solid lines in show the median of relative fitness versus the iteration number. For large privacy budgets, the median decreases across time until the algorithm converges to a neighbourhood of the solution of (1). The relative fitness of the trained model also improves as $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases.

Figure 10 shows the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The

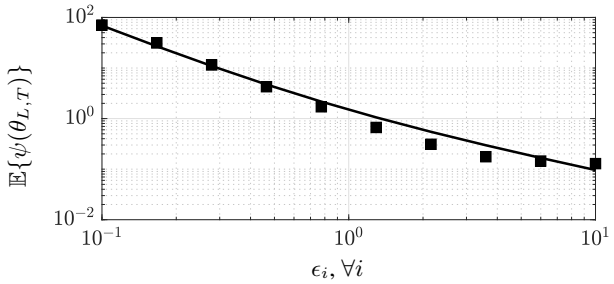

Figure 10: Relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations versus the privacy budget ϵ_i, $\forall i$. The solid line illustrates the bound in (6) with $\bar{c}_1' = 0.9$ and $\bar{c}_2' = 0.6$.

markers (i.e., ■) show the experiments. Evidently, the relative fitness rapidly improves as the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increase.

Figure 11 illustrates the relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, for three choices of privacy budgets $\epsilon_i = 0.1$ (black line), $\epsilon_i = 1$ (dashed line), $\epsilon_i = 10$ (dash-dotted line). The markers show the relative fitness of the non-private ML model θ_i^*, $\psi(\theta_i^*)$, constructed based on only the private data of the i-th data owner versus the size of the data set owned by the i-th data owner. For $\epsilon = 10$, eight hospitals (i.e., Women And Children's Hospital Of Buffalo, Crouse Hospital, St Peters Hospital, White Plains Hospital Center, Westchester Medical Center, Memorial Hospital for Cancer and Allied Diseases, Long Island Jewish Schneiders Children's Hospital Division, St Francis Hospital) benefit from collaboration. The relative fitnesses of the non-private ML model θ_i^* for these eight hospitals are above the dash-dotted line.

Discussions, Conclusions, and Future Research

In this paper, we developed an asynchronous DP algorithm for training ML models on multiple private datasets. We proved that, by following the asynchronous algorithm in this paper, the cost of privacy is inversely proportional to the combined size of the training datasets squared and the privacy budgets squared. Finally, we validated the theoretical results on experiments on financial data. Future work can focus on:

- An interesting extension is to consider multiple learners training separate ML models. This would be more similar to the distributed ML on arbitrary connected graphs. This way, we can extend the results of this paper to more general communication structures with the learner not necessarily at the center.

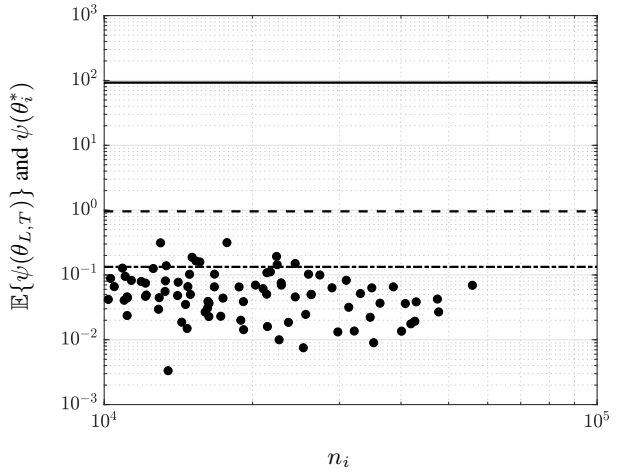

Figure 11: Relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, for three choices of privacy budgets $\epsilon_i = 0.1$ (black line), $\epsilon_i = 1$ (dashed line), $\epsilon_i = 10$ (dash-dotted line). The markers show the relative fitness of the non-private ML model θ_i^*, $\psi(\theta_i^*)$, constructed based on only the private data of the i-th data owner versus the size of the data set owned by the i-th data owner. For $\epsilon = 10$, eight hospitals (i.e., Women And Children's Hospital Of Buffalo, Crouse Hospital, St Peters Hospital, White Plains Hospital Center, Westchester Medical Center, Memorial Hospital for Cancer and Allied Diseases, Long Island Jewish Schneiders Children's Hospital Division, St Francis Hospital) benefit from collaboration. The relative fitnesses of the non-private ML model θ_i^* for these eight hospitals are above the dash-dotted line.

- We can investigate the behaviour of private data owners and learners in a data market. The cost of privacy in this paper can be used as a guide for developing compensation mechanisms for private data owners to increase their privacy budgets. The developed algorithm is particularly of use as the data owners and the learners in the data market can predict the performance of privately-trained ML models during negotiation for setting privacy budgets and compensating data owners.
- Finally, we can extend the framework of this paper to adversarial ML with data owners and learners that are more sophisticated than curious-but-honest adversaries in this paper.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security* 13(5):1333–1345.

Aybat, N.; Wang, Z.; and Iyengar, G. 2015. An asynchronous distributed proximal gradient method for composite convex optimization. In *International Conference on Machine Learning*, 2454–2462.

Bassily, R.; Smith, A.; and Thakurta, A. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 464–473. IEEE.

Bedi, A. S., and Rajawat, K. 2018. Asynchronous incremental stochastic dual descent algorithm for network resource allocation. *IEEE Transactions on Signal Processing* 66(9):2229–2244.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191. ACM.

Chaudhuri, K., and Monteleoni, C. 2009. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, 289–296.

Du, W.; Han, Y. S.; and Chen, S. 2004. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, 222–233. SIAM.

Farokhi, F., and Johansson, K. H. 2014. Stochastic sensor scheduling for networked control systems. *IEEE Transactions on Automatic Control* 59(5):1147–1162.

Gilad-Bachrach, R.; Dowlin, N.; Laine, K.; Lauter, K.; Naehrig, M.; and Wernsing, J. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, 201–210.

Graepel, T.; Lauter, K.; and Naehrig, M. 2012. ML confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology*, 1–21. Springer.

Heidelberger, P., and Trivedi, K. S. 1982. Queueing network models for parallel processing with asynchronous tasks. *IEEE Transactions on Computers* (11):1099–1109.

Hong, M., and Chang, T.-H. 2017. Stochastic proximal gradient consensus over random networks. *IEEE Transactions on Signal Processing* 65(11):2933–2948.

Huang, Z.; Hu, R.; Gong, Y.; and Chan-Tin, E. 2018. DP-ADMM: ADMM-based distributed learning with differential privacy. *Preprint: arXiv preprint arXiv:1808.10101*.

Hunt, T.; Song, C.; Shokri, R.; Shmatikov, V.; and Witchel, E. 2018. Chiron: Privacy-preserving machine learning as a service. *arXiv preprint arXiv:1803.05961*.

Jagannathan, G., and Wright, R. N. 2005. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 593–599. ACM.

Kaggle. 2018. Lending club loan data: Analyze lend-

ing club's issued loans. https://www.kaggle.com/wendykan/lending-club-loan-data, Date Accessed: 17 Oct 2018.

Lagunoff, R., and Matsui, A. 1997. Asynchronous choice in repeated coordination games. *Econometrica* 1467–1477.

Li, P.; Li, J.; Huang, Z.; Li, T.; Gao, C.-Z.; Yiu, S.-M.; and Chen, K. 2017. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems* 74:76–85.

Lindell, Y., and Pinkas, B. 2000. Privacy preserving data mining. In Bellare, M., ed., *Advances in Cryptology — CRYPTO 2000*, 36–54. Springer Berlin Heidelberg.

McMahan, B., and Streeter, M. 2014. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, 2915–2923.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2017. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 1928–1937.

New York State Department of Health. 2015. Hospital Inpatient Discharges (SPARCS De-Identified): 2015. https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/82xm-y6g8, Date Accessed: 20 July 2019.

Parkes, D. C., and Shneidman, J. 2004. Distributed implementations of vickrey-clarke-groves mechanisms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 261–268. IEEE Computer Society.

Ram, S. S.; Nedić, A.; and Veeravalli, V. V. 2009. Asynchronous gossip algorithms for stochastic optimization. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 3581–3586. IEEE.

Sarwate, A. D., and Chaudhuri, K. 2013. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Processing Magazine* 30(5):86–94.

Shokri, R., and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1310–1321. ACM.

Smyth, P.; Welling, M.; and Asuncion, A. U. 2009. Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, 81–88.

Srivastava, K., and Nedic, A. 2011. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing* 5(4):772–790.

Tanaka, T.; Farokhi, F.; and Langbort, C. 2015. Faithful implementations of distributed algorithms and control laws. *IEEE Transactions on Control of Network Systems* 4(2):191–201.

Touri, B.; Nedič, A.; and Ram, S. S. 2010. Asynchronous stochastic convex optimization over random networks: Error bounds. In *2010 Information Theory and Applications Workshop (ITA)*, 1–10.

Vaidya, J., and Clifton, C. 2002. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 639–644. ACM.

Vaidya, J.; Kantarcıoğlu, M.; and Clifton, C. 2008. Privacy-preserving naive bayes classification. *The VLDB Journal* 17(4):879–898.

Wu, N.; Farokhi, F.; Smith, D.; and Kaafar, M. A. 2020. The value of collaboration in convex machine learning with differential privacy. In *Proceedings of the 41st IEEE Symposium on Security and Privacy*.

Zhang, T., and Zhu, Q. 2017. Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security* 12(1):172–187.

Zhang, J.; Zhang, Z.; Xiao, X.; Yang, Y.; and Winslett, M. 2012. Functional mechanism: Regression analysis under differential privacy. *Proceedings of the VLDB Endowment* 5(11):1364–1375.

Zhang, T.; He, Z.; and Lee, R. B. 2018. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860*.

Zhang, Z.; Rubinstein, B. I. P.; and Dimitrakakis, C. 2016. On the differential privacy of Bayesian inference. In *AAAI Conference on Artificial Intelligence*, 2365–2371.

Proof of Theorem 1

Since there are at most T rounds of communication, the privacy budget in each step should be set as ϵ_i/T for all i. Now, note that

$$\begin{aligned}
&\|\mathcal{Q}_{i_k}(\mathcal{D}_{i_k};\bar{\theta}_k)-\mathcal{Q}_{i_k}(\mathcal{D}'_{i_k};\bar{\theta}_k)\|_1\\
&\qquad=\frac{1}{n_{i_k}}\Bigg\|\sum_{\{x,y\}\in\mathcal{D}_{i_k}}\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)\\
&\qquad\qquad-\sum_{\{x,y\}\in\mathcal{D}'_{i_k}}\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)\Bigg\|_1\\
&\qquad=\frac{1}{n_{i_k}}\Big\|\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)|_{\{x,y\}\in\mathcal{D}_{i_k}\setminus\mathcal{D}'_{i_k}}\\
&\qquad\qquad-\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)|_{\{x,y\}\in\mathcal{D}'_{i_k}\setminus\mathcal{D}_{i_k}}\Big\|_1\\
&\qquad=\frac{2\Xi}{n_{i_k}}.
\end{aligned}$$

Therefore, the scale of the noise must be selected as $2\Xi T/(n_{i_k}\epsilon_{i_k})$.

Proof of Theorem 2

We start by casting the problem of privacy-aware learning in the framework of asynchronuous disributed optimization

in (Touri, Nedič, and Ram 2010). For any $\eta < 1/N$, we can define

$$f_i(\theta) = \eta g(\theta) + \frac{1}{n} \sum_{\{x,y\}\in\mathcal{D}_i} \ell(\mathfrak{M}(x;\theta), y), \forall i \in \mathcal{N},$$

and

$$f_L(\theta) = (1 - \eta N) g(\theta).$$

We can think of f_i as the cost functions of data owners and f_L as the cost function of the learner. By construct, f_L is σ_L strongly convex with $\sigma_L = (1-\eta N)\sigma$ and f_i is σ_i strongly convex with $\sigma_i = \eta\sigma$. Note that

$$\begin{aligned}\|\nabla_\theta f_i(\theta)\|_2 =& \left\| \eta \nabla_\theta g(\theta) + \frac{1}{n} \sum_{\{x,y\}\in\mathcal{D}_i} \nabla_\theta \ell(\mathfrak{M}(x;\theta), y) \right\| \\ \leq& \eta \Xi_g + \frac{n_i}{n} \Xi \\ \leq& \Xi_g + \Xi,\end{aligned}$$

and

$$\begin{aligned}\|\nabla_\theta f_L(\theta)\|_2 =& \|(1-\eta N)\nabla_\theta g(\theta)\| \\ \leq& (1-\eta N)\Xi_g \\ \leq& \Xi_g.\end{aligned}$$

Therefore, $\|\nabla_\theta f_i(\theta)\|_2 \leq C, \forall i$, and $\|\nabla_\theta f_L(\theta)\|_2 \leq C$ with $C = \Xi_g + \Xi$.

In each iteration, one of the data owners at random is selected and follows the gossip algorithm (see (Touri, Nedič, and Ram 2010)) for exchanging information in learning and updating the decision variables. In this paper, however, we assume that the learner takes care of all the updates and storing the iterates. Therefore, the learner submits a gradient query to the selected data owner and receives a DP response for updating the decision variables. Let i denote the index of the randomly-selected data owner at iteration k; note that i_k is used in Algorithm 1 for denoting the index. We use $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote a graph with the vertex set $\mathcal{V} = \{1, \ldots, N, N+1\}$, in which node $N+1$ is the learner L, and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. By the methodology of (Touri, Nedič, and Ram 2010), we get

$$W_k = I - \frac{1}{2}(e_i - e_{N+1})(e_i - e_{N+1})^\top,$$

and $U_k = \{L, i\}$. It is evident that the probability of selecting the learner at each round is equal to one, i.e., $\gamma_L = 1$, and the probability of selecting any data owner is $\gamma_i = 1/N$ in the notation of (Touri, Nedič, and Ram 2010),. We get

$$\begin{aligned}\overline{W} =& \mathbb{E}\{W_k\} \\ =& I - \begin{bmatrix} \frac{1}{2N} & 0 & \cdots & 0 & -\frac{1}{2N} \\ 0 & \frac{1}{2N} & \cdots & 0 & -\frac{1}{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2N} & \frac{1}{2N} \\ -\frac{1}{2N} & -\frac{1}{2N} & \cdots & -\frac{1}{2N} & \frac{1}{2} \end{bmatrix}.\end{aligned}$$

We meet all the conditions of Assumption 2 in (Touri, Nedič, and Ram 2010). Furthermore, using Theorem 1 in (Touri, Nedič, and Ram 2010), we can see that

$$\lambda = \left\| W_k - \frac{1}{N+1} \mathbb{1}\mathbb{1}^\top W_k \right\|_2^2 < 1.$$

The updates in (2) in (Touri, Nedič, and Ram 2010) can be rewritten as

$$\bar{\theta}_k = \frac{1}{2}\theta_{L,k-1} + \frac{1}{2}\theta_{i,k-1},$$

with the notation substitution of $\bar{\theta}_k$ instead of $v_{i,k} = v_{L,k}$, $\theta_{i,k}$ instead of $x_{i,k}$, and $\theta_{L,k}$ instead of $x_{L,k}$. The updates in (3) in (Touri, Nedič, and Ram 2010) can also be rewritten as

$$\begin{aligned}\theta_{i,k} =& \Pi_\Theta \left[\bar{\theta}_k - \alpha_i \eta \nabla_\theta g(v_k) + \alpha_i \frac{n_i}{n} \overline{\mathcal{Q}}_i(\bar{\theta}_k; k) \right] \\ =& \Pi_\Theta \left[\bar{\theta}_k - \alpha_i \left(\eta \nabla_\theta g(v_k) + \frac{n_i}{n} \left(\mathcal{Q}_i(\bar{\theta}_k; k) + n_{i,k} \right) \right) \right] \\ =& \Pi_\Theta \left[\bar{\theta}_k - \alpha_i (\nabla_\theta f_i(\bar{\theta}_k) + w_i(k)) \right],\end{aligned}$$

and

$$\begin{aligned}\theta_{L,k} =& \Pi_\Theta \left[\bar{\theta}_k - \alpha_L \nabla f_L(\bar{\theta}_k) \right] \\ =& \Pi_\Theta \left[\bar{\theta}_k - (1-\eta N)\alpha_L \nabla_\theta g(\bar{\theta}_k) \right],\end{aligned}$$

where $w_i(k)$ is the additive i.i.d. DP noise and

$$\mathcal{Q}_i(\bar{\theta}_k; k) = \frac{1}{n_i} \sum_{\{x,y\}\in\mathcal{D}_i} \nabla_\theta \ell(\mathfrak{M}(x; \bar{\theta}_k), y).$$

We have

$$\begin{aligned}\mathbb{E}\{n_{i,k}|\mathcal{F}_k\} &= 0, \\ \mathbb{E}\{\|n_{i,k}\|_2^2|\mathcal{F}_k\} &\leq \nu_i^2,\end{aligned}$$

where $\mathcal{F}_k$ is the filtration generated by the entire history of Algorithm 1 up to iteration k. Using Theorem 1, we can see that

$$\nu_i = \frac{2\sqrt{2}\Xi T}{n\epsilon_i}.$$

Extending Lemma 3 in (Touri, Nedič, and Ram 2010) results in

$$\begin{aligned}\mathbb{E}\{\|\nabla_\theta f_i(\bar{\theta}_k) + w_i(k)\|_2^2|\mathcal{F}_{k-1}, W_k\} &\leq C^2 + \nu_i^2 \\ &\leq (C + \nu_i)^2, \\ \mathbb{E}\{\|\nabla_\theta f_L(\bar{\theta}_k)\|_2^2|\mathcal{F}_{k-1}, W_k\} &\leq C^2.\end{aligned}$$

Therefore, we can upgrade the right-hand side of (22) in (Touri, Nedič, and Ram 2010) to

$$\mathbb{E}\{\alpha_i^2 (C+\nu_i)^2\} + \alpha_L^2 C^2 = \alpha_L^2 C^2 + \frac{1}{N} \sum_{i\in\mathcal{N}} \alpha_i^2 (C+\nu_i)^2$$

Note that, in the case of this paper, the summation only contains two terms because, in each iteration, only the learner and another data owner update their decision variables. This implies that, in Proposition 1 in (Touri, Nedič, and Ram 2010), ε_{net} must be updated to

$$\varepsilon_{\text{net}} = \frac{C\sqrt{N+1}}{1-\sqrt{\lambda}} \sqrt{\alpha_L^2 + \frac{1}{N} \sum_{i\in\mathcal{N}} \alpha_i^2 \left(1 + \frac{\nu_i}{C}\right)^2}.$$

With the same line of reasoning, we can improve the bound in Proposition 2 in (Touri, Nedič, and Ram 2010) to get

$$\limsup_{k\to\infty}\left[\mathbb{E}\{\|\theta_{L,k}-\theta^*\|_2^2\}+\sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,k}-\theta^*\|_2^2\}\right] \leq \frac{\varepsilon+2\alpha_{\max}C\varepsilon_{\text{net}}}{1-q}, \quad (7)$$

where

$$\begin{aligned}\varepsilon =&2(N+1)(1-\gamma_{\min})\delta_{\alpha,\sigma}\operatorname{diam}(\Theta)^2\\ &2(N+1)\delta_{\alpha,\gamma}C\operatorname{diam}(\Theta)\\ &+C^2\left(\alpha_L^2+\frac{1}{N}\sum_{i\in\mathcal{N}}\alpha_i^2\left(1+\frac{\nu_i}{C}\right)^2\right),\end{aligned} \quad (8)$$

and

$$\begin{aligned}\alpha_{\max} =&\max_i \alpha_i,\\ \gamma_{\min} =&1/N,\\ q =&1-2\gamma_{\min}\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}\\ \delta_{\alpha,\sigma} =&\max\left\{\alpha_L\sigma_L,\max_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}-\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}\\ \delta_{\alpha,\gamma} =&\max\left\{\alpha_L\gamma_L,\max_{i\in\mathcal{N}}\alpha_i\gamma_i\right\}-\min\left\{\alpha_L\gamma_L,\min_{i\in\mathcal{N}}\alpha_i\gamma_i\right\}.\end{aligned}$$

Therefore, for any $\varsigma>0$, there exists large enough $T\in\mathbb{N}$ such that

$$\left[\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}+\sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,T}-\theta^*\|_2^2\}\right] \leq \varsigma+\frac{\varepsilon+2\alpha_{\max}C\varepsilon_{\text{net}}}{1-q}, \quad (9)$$

Selecting $\eta=1/(2N)$ and $\alpha_L=\alpha_i/N=\alpha/\sigma$ for some constant $\alpha\in(0,1)$, we get $\delta_{\alpha,\sigma}=\delta_{\alpha,\gamma}=0$. Therefore, we can simplify (8) to get

$$\varepsilon=\frac{\alpha^2C^2}{\sigma^2}\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{\nu_i}{C}\right)^2\right) \quad (10)$$

We will also get

$$\begin{aligned}2\alpha_{\max}C\varepsilon_{\text{net}}=&\frac{2N\alpha^2C^2\sqrt{N+1}}{\sigma^2(1-\sqrt{\lambda})}\\ &\times\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{\nu_i}{C}\right)^2}.\end{aligned} \quad (11)$$

Furthermore,

$$1-q=2\gamma_{\min}\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}=\frac{\alpha}{N}. \quad (12)$$

Combining (9) with (10)–(12), we get

$$\begin{aligned}&\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}\\ \leq&\left[\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}+\sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,T}-\theta^*\|_2^2\}\right]\\ \leq&\frac{N\alpha C^2}{\sigma^2}\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2\right)\\ &+\frac{2N^2\alpha C^2\sqrt{N+1}}{\sigma^2(1-\sqrt{\lambda})}\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2}.\end{aligned}$$

Define

$$c_1=\frac{NC^2}{\sigma^2},\quad c_2=\frac{2N^2C^2\sqrt{N+1}}{\sigma^2(1-\sqrt{\lambda})}.$$

We have

$$\begin{aligned}&\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}\\ &\leq c_1\alpha\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2\right)\\ &\quad+c_2\alpha\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2}.\end{aligned}$$

Selecting $\alpha=\rho/T^2$ and noting that $\Xi\leq\Xi_g+\Xi$, the upper bound can be further simplified (3). Following the same modifications in the proof of Proposition 3 in (Touri, Nedič, and Ram 2010) results in (4).