Assisted Learning: A Framework for Multi-Organization Learning

Xun Xian
School of Statistics
University of Minnesota
xian0044@umn.edu

Xinran Wang
School of Statistics
University of Minnesota
wang8740@umn.edu

Jie Ding
School of Statistics
University of Minnesota
dingj@umn.edu

Reza Ghanadan
Google LLC
Washington DC
rezaghanadan@google.com

Abstract

In an emerging number of AI tasks, collaborations among different organizations or agents (e.g., human and robots, mobile units) are often essential to accomplish an agent-specific mission. However, to avoid leaking useful and possibly proprietary information, organizations typically enforce stringent security constraints on sharing both models (or algorithms) and data, which significantly limits collaborations. In this work, we introduce the Assisted Learning framework for agents to assist each other in supervised learning tasks without transmitting any agent's private model, objective, or data. Bob assists Alice either by building a predictive model using Alice's numeric labels or improving Alice's learning through iterative transmissions of task-specific statistics. Theoretical and experimental studies, including real-world medical benchmarks, show that Assisted Learning can often achieve near-oracle learning performance as if models and data were centralized. The framework of Assisted Learning promises a systemic and autonomous acquisition of a diverse body of information broadly exploited by decentralized agents.

1 Introduction

One of the significant characteristics of big data is variety, featuring in a large number of statistical learners, each with personalized data and domain-specific learning objectives. While building an entire database by integrating all the accessible data sources could provide an ideal dataset for learning, sharing heterogeneous data among multiple organizations typically leads to tradeoffs between learning efficiency and data privacy. As the awareness of privacy arise with a decentralized set of organizations(learners), a realistic challenge is to protect learners' privacy with respect to data as well as sophisticated models.

There exists a large market of collaborative learning. We'll take the medical industry as an example to illustrate such learning scenarios and their pros and cons. It is common for medical organizations to acquire others' assistance in order to improve clinical care [1], reduce capital costs [2] , and accelerate scientific progress [3]. Consider two organizations Alice (a hospital) and Bob (a related clinical laboratory) who collect various features from the same group of people. Now Alice wants to predict the Length of Stay (LOS), which is one of the most important driving forces of hospital costs [4].

Scenario 1: If the organization doesn't have the ability of modeling or computing, then it has to sacrifice the data privacy for the assistance of learning. In our example, if Alice doesn't have much

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Table 1: Examples of Bob assisting Alice (none of whom will transmit personalized models or data).

Alice	Hospital	Mobile device	Investor	EEG
Bob	Clinical Lab	Cloud service	Financial trader	Eye-movement
Collating Index	Patient ID	User ID	Time stamp	Subject ID

knowledge of machine learning or enough computational resources, then she may prefer to be assisted by Bob via Machine-Learning-as-a-Service (MLaaS) [5, 6]. In MLaaS, a service provider Bob receives predictor-label pairs (x, y) from Alice, and then learns a private supervised model. After the learning stage, Bob provides prediction services upon Alice's future data by applying the learned model. In this learning scenario, Alice gains information from Bob's modeling, but at the cost of exposing her private data.

Scenario 2: If the organization has the ability of modeling and computing, then it often brings information loss or pays higher communication/computation costs when assisted by other organizations. Suppose that Alice is proficient in machine learning and has computational resources, how can she benefit from Bob's relevant medical data (model)? One way for Alice to privatize data is to add noises and then transmit privatized data. Though this strategy has provable privacy guarantees (as evaluated under the framework of differential privacy (DP) [7–9], it often leads to information loss and thus degraded the learning performance. Another solution for Alice is to send data with homomorphic encryption (HE) [10]. While it is information lossless, it may suffer from intractable communication/computation costs.

Privacy sensitive organizations from various industries will not or cannot transmit their personalized models or data. Some common examples of such organizations are listed in Tab. 1. Under this limit, for the above two scenarios, is it possible for Bob to assist Alice *without* data sharing?

For Scenario 1, Bob could choose to simply receive Alice's id-label pairs (if not private), collate them with his own private data, and learn a supervised model privately. Bob then provides prediction services for Alice who inquires with future data, for example in the form of an application programming interface (API). For Scenario 2, suppose that Alice also has a private learning model and private data features that can be (partially) collated to Bob's. Is it possible to still leverage the *model* as well as *data* held by Bob? A classical approach for Alice is to perform model selection from her own model and Bob's private model (through Bob's API), and then decide whether to use Bob's service in the future. A related approach is to perform statistical model averaging over the two learners. However, neither approach will significantly outperform the better one of Alice's and Bob's [11, 12]. This is mainly because that model selection or averaging in the above scenario fails to fully utilize all the available data, which is a union of Alice's and Bob's.

Is it possible for Alice to achieve the performance as if all the private information of Alice and Bob were centralized (so that the 'oracle performance' can be obtained)? This motivates us to propose the framework of *Assisted Learning*, where the main idea is to treat predictors $\boldsymbol{x}$ as private and use a suitable choice of 'y' at each round of assistance, so that Alice may benefit from Bob as if she had Bob's data.

The main contributions of this work are threefold: First, we introduce the notion of Assisted Learning which is naturally suitable for contemporary machine learning markets. Second, in the context of Assisted Learning, we develop two concrete protocols so that a service provider can assist others by improving their predictive performance, without the need for central coordination. Third, we show that the proposed learning protocol can be applied to a wide range of nonlinear and nonparametric learning tasks, where near-oracle performance can be achieved.

The rest of the paper is organized as follows. First, we briefly discuss the recent development of some related areas in Section 2. Then, a real-world application is presented to demonstrate assisted learning's reasonability and necessity in Section 3. In Section 4, we formally introduce assisted learning and give theoretical analysis. In Section 5, we provide experimental studies on both real and synthetic datasets. We conclude in Section 6.

2 Related work

There has been a lot of existing research that considers distributed data with heterogeneous features (vertically partitioned/split data) for the purpose of collaborative learning. Early work on privacy-preserving learning on vertically partitioned data (e.g. [13, 14]) need to disclose certain features' class distributions, which may lead to privacy breach. Recent advancement in decentralized learning is Federated Learning [15–17], where the main idea is to learn a joint model using data that are distributed among participants (e.g., mobile devices) without sharing data directly . The data scenario (i.e., participants hold different features over the same group of subjects) in vertical federated learning is similar to that in assisted learning. Nevertheless, these two types of learning are fundamentally different in that vertical federated learning aims to construct a joint global model with each participant holding a part of global model parameters, whereas assisted learning focuses on acquiring useful side information from others learners, without using a prescribed learning model.

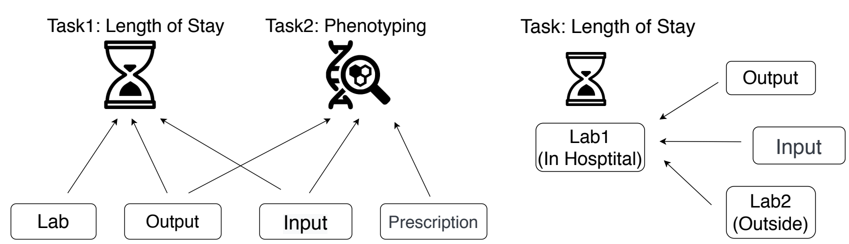
Vertical Federated Learning. Particular methods of Vertical Federated Learning include those based on homomorphic encryption, and/or stochastic gradient descent on partial parameters to jointly optimize a global model [18–21]. In all of these schemes, the system needs to be synchronized (among all the participants) in the training process. In contrast, assisted learning is model-free meaning that the models of each participant can be arbitrary, and thus it does not require synchronized updating or the technique of homomorphic encryption.

Secure Multi-party Computation. Another related literature is Secure Multi-party Computation [22, 23] where the main idea is to securely compute a function in such a way that no players can learn anything more than its prescribed output. Several work under this framework studied machine learning on vertically partitioned data [24–26], and they typically rely on an external party. The use of external service may give rise to trust-related issues. Assisted learning does not require third-party service, and participants often assist each other by playing the roles of both service provider and learner (see Section 4.2).

Multimodal Data Fusion. Data Fusion [27–29] aims to aggregate information from multiple modalities to perform a prediction. Its focus is to effectively integrate complementary information and eliminate redundant information, often assuming that data are already centralized by one learner. In assisted learning, the goal is to develop cooperative learning for multiple organizations/learners without data sharing.

3 Real-world Applications of Assisted Learning on MIMIC3 Benchmarks

Medical Information Mart for Intensive Care III [30] (MIMIC3) is a comprehensive clinical database that contains de-identified information for 38,597 distinct adult patients admitted to critical care units between 2001 and 2012 at the Beth Israel Deaconess Medical Center in Boston, Massachusetts. Data in MIMIC3 are stored in 26 tables, which can be linked by unique admission identifiers. Each table corresponds to the data 'generated' from a certain source. For example, the *LAB* table consists of patients' laboratory measurements, the *OUTPUT* table includes output information from patients, and the *PRESCRIPTIONS* table contains medications records.



(a) Each rectangle represents one specific data table in MIMIC3. For the task of predicting the Length of Stay, the data from *Lab*, *Output*, and *Input* are used. For the task of classifying Phenotyping, the data from *Output*, *Input*, and *Prescription* are used.

(b) Assisted Learning for MIMIC3. Lab1 wants to predict the Length of Stay. Three other data tables, i.e., *Output*, *Input* and *Outside Lab* provide assistance.

Figure 1: Illustrations of (a) MIMIC3 Benchmarks structure and (b) assisted learning for MIMIC3 Benchmarks.

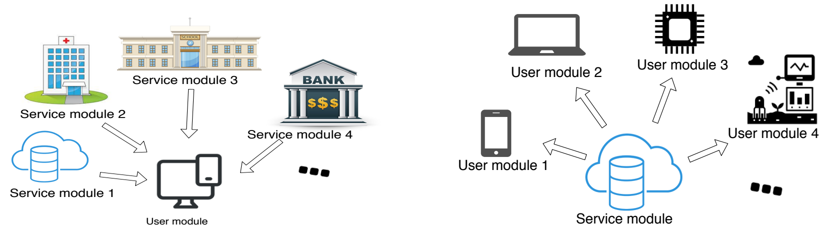
MIMIC3 Benchmarks [31, 32] consist of essential medical machine learning tasks, e.g., predicting the Length of Stay (LOS) in order to manage resource and measure patient's acuity, and classifying Phenotype for scientific research. Commonly, every single task involves medical information from different data tables as depicted in Fig. 1a. For example, for the in-hospital Lab (Lab1) to predict LOS, a usual approach is to first centralize data from *Lab*, *Output*, and *Input*, and then construct a model on the joint database. However, sharing sensitive medical data may not be allowed for both data contributor (e.g. patient) and data curator (e.g. hospital), and therefore a method that achieves the best possible learning performance without *sharing* the data is in urgent demand. To summarize, this application domain involves multiple organizations (namely data tables/divisions) with discordant learning goals as well as heterogeneous/multimodal data whose sharing is prohibited. For Lab1 to predict LOS via assisted learning in the presence of Output, Input, and Lab2, the main idea is for those learners to assist Lab 1 by iteratively transmitting only task-relevant statistics instead of raw data. A reasonable evaluation metric of the assisted learning performance is to compare it with the performance produced by centralizing data.

4 Assisted Learning

Throughout the paper, we let $X \in \mathcal{X}^{n \times p}$ denote a general data matrix which consists of n items and p features, and $y \in \mathcal{Y}^n$ be a vector of labels (or responses), where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}$. Let x_i denotes the ith row of X. A supervised function f approximates $x_i \mapsto \mathbb{E}(y_i \mid x_i)$ for a pair of predictor (or feature) $x_i \in \mathcal{X}^p$ and $y_i \in \mathcal{Y}$. Let $f(X)$ denote an $\mathbb{R}^n$ vector whose ith element is $f(x_i)$. We say two matrices or column vectors A, B are *collated* if rows of A and B are aligned with some common index. For example, the index can be date or time stamps for datasets of time series, or personal identification number for datasets of mobile users.

4.1 General Description of Assisted Learning

We first depict how we envision Assisted Learning through a concrete usage scenario based on MIMIC3 Benchmark. Alice (Intensive Care Unit) is equipped with a set of labeled data (X_A, Y_A) and supervised learning algorithms. And m other divisions may be performing different learning tasks with distinct data $(X_i, Y_i)_{i=1,2,\ldots,m}$ and learning models, where $(X_i)_{i=1,2,\ldots,m}$ can be (partially) collated. Alice wishes to be assisted by others to facilitate her own learning while maintaining both of their sensitive information. On the other hand, Alice would also be glad to assist others for potential rewards. A set of learning modules such as Alice constitute a statistical learning market where each module can either provide or receive assistance to facilitate personalized learning goals. In the following, we introduce our notion of algorithm and module in the context of supervised learning. Figure 2 illustrates Assisted Learning from a user's perspective and a service provider's perspective.



(a) User-centered perspective: a user module sends queries to service modules to obtain feedbacks and improve its personal learning.

(b) Service-centered perspective: a service module assists user modules by delivering feedbacks using its personal data and model.

Figure 2: Assisted Learning from two perspectives.

Definition 1 (Algorithm). *A learning algorithm $\mathcal{A}$ is a mapping from a dateset $X \in \mathbb{R}^{n \times p}$ and label vector $y \in \mathbb{R}^n$ to a prediction function $f_{\mathcal{A},X,y} : \mathbb{R}^p \to \mathbb{R}$.*

An algorithm may represent linear regression, ensemble method, neural networks, or a set of models from which a suitable one is chosen using model selection techniques [12, 33]. For example, when the least squares method is used to learn the supervised relation between X and y, then $f_{\mathcal{A},X,y}$ is a linear operator: $\tilde{x} \mapsto \tilde{x}^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ for a predictor $\tilde{x} \in \mathbb{R}^p$. The above $f_{\mathcal{A},X,y}$ is also called a hypothesis in some literature of classification.

Definition 2 (Module). *A module $\mathcal{M} = (\mathcal{A}, X)$ is a pair of algorithm $\mathcal{A}$ and observed dateset X. For a given label vector $y \in \mathbb{R}^n$, a module naturally induces a prediction function $f_{\mathcal{A},X,y}$. We simply write $f_{\mathcal{A},X,y}$ as $f_{\mathcal{M},y}$ whenever there is no ambiguity.*

Concerning MIMIC3, a division with its data table and machine learning algorithm is a module. In the context of assisted learning, $\mathcal{M} = (\mathcal{A}, X)$ is treated as private and y is public. If y is from a benign user Alice, it represents a particular task of interest. The prediction function $f_{\mathcal{M},y} : \mathcal{X}^p \to \mathcal{Y}$ is thus regarded as a particular model learned by $\mathcal{M}$ (Bob), driven by y, in order to provide assistance. Typically $f_{\mathcal{M},y}$ is also treated as private.

Definition 3 (Assisted Learning System). *An assisted learning system consists of a module M, a learning protocol, a prediction protocol, and the following two-stage procedure.*

- *In stage I ('learning protocol'), module $\mathcal{M}$ receives a user's query of a label vector $y \in \mathcal{Y}^n$ that is collated with the rows of X; a prediction function $f_{\mathcal{M},y}$ is produced and privately stored; the fitted value $f_{\mathcal{M},y}(X) = [f_{\mathcal{M},y}(x_1), \ldots, f_{\mathcal{M},y}(x_n)]^{\mathrm{T}}$ is sent to the user.*
- *In stage II ('prediction protocol'), module $\mathcal{M}$ receives a query of future predictor $\tilde{x}$; its corresponding prediction $\hat{y} = f_{\mathcal{M},y}(\tilde{x})$ is calculated and returned to the user.*

In the above Stage I, the fitted value, $f_{\mathcal{M},y}(X)$, returned from the service module (Bob) upon an inquiry of y, is supposed to inform the user module (Alice) of the training error so that Alice can take subsequent actions. Bob's actual predictive performance is reflected in Stage II. The querying user in Stage II may or may not be the same user as in stage I.

4.2 A Specific Learning Scenario: Iterative Assistance

Suppose that Alice not only has a specific learning goal (labels) but also has private predictors and algorithm, how could Alice benefit from other modules/learners through the two stages of assisted learning? We address this by developing a specific user scenario of assisted learning, where we consider regression methods. For Alice to receive assistance from other modules, their data should be at least partially collated. For brevity, we assume that the data of all the modules can be collated using public indices. Procedure 1 outlines a realization of assisted learning between Alice with m other modules. The main idea is to let Alice seek assistance from various other modules through iterative communications where only few key statistics are transmitted.

The key idea is to allow each module to only transmit fitted residuals to the other module, iterating until the learning loss is reasonably small. In the *learning stage* (Stage I), at each round of assistance k, Alice first sends a query to each module $\mathcal{M}_j$ by transmitting its latest statistics $e_{j,k}$; upon receipt of the query, if module j agrees, it treats $e_{j,k}$ as labels and fit a model $\hat{A}_{j,k}$ (based on the data aligned with such labels); module j then fits residual $\tilde{e}_{j,k}$ and sends it back to module Alice. Alice processes the collected responses $\tilde{e}_{j,k}, \ldots$ ($j = 1, \ldots, m$), and initializes the $k+1$ round of communication/assistance. After the above procedure stops at an appropriate stopping time $k = K$, the training stage for Alice is suspended. In the *prediction stage* (Stage II), upon arrival of a new feature vector x^*, Alice queries the prediction results $\hat{A}_{j,k}(x^*)$ ($k = 1, 2, \ldots, K$) from module j, and combine them to form the final prediction $\tilde{y}^*$. Several ways to combine predictions from other modules are discussed in Remark 1. Here, we use unweighted summation as the default method to combine predictions. A simple illustration of the two-stage procedures for Bob to assist Alice is depicted in Fig. 3.

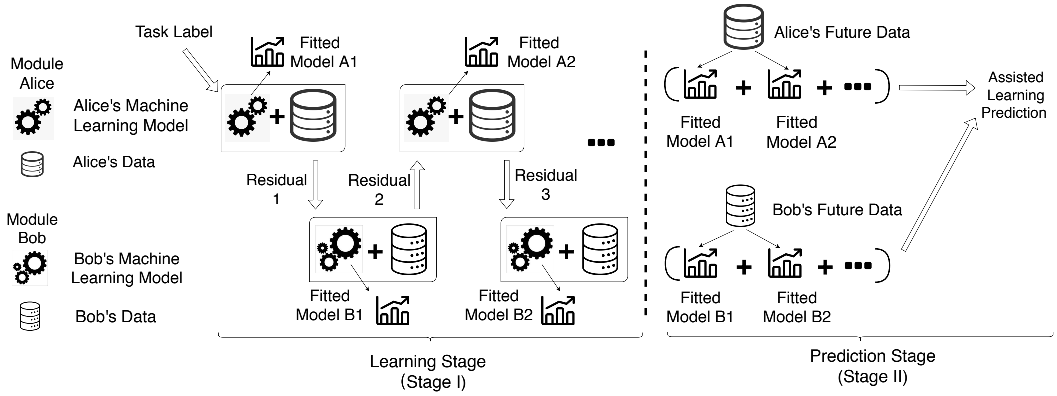


Figure 3: Illustration of the two-stage procedures for Bob to assist Alice.

It is natural to consider the following oracle performance without any privacy constraint as the limit of learning. Let ℓ denote some loss function (e.g. squared loss for regression). The *oracle performance* of learner Alice $\mathcal{M}_0 = (\mathcal{A}_0, X_0)$ in the presence of module $\mathcal{M}_j = (\mathcal{A}_j, X_j)_{j=1,2,\ldots,m}$ is defined by $\min_{\mathcal{A}_j} \mathbb{E}\{\ell(y^*, f_j(x^*))\}$ where the prediction function f_j is learned from algorithm $\mathcal{A}_j$ using all the collated data $\bigcup_{i=0}^{m} X_j$. In other words, it is the optimal out-sample loss produced from the candidate methods and the pulled data of all modules. The above quantity provides a theoretical limit on what assisted learning can bring to module Alice. Under some conditions, we show that Alice will approach the oracle performance through assisted learning with Bob, without direct access to Bob's data X_B. In the experimental section, an interesting observation is also presented, which is a tradeoff between the rounds of assistance and learning performance that strikingly resembles the classical tradeoff between model complexity and learning performance.

Theorem 1. Suppose that Alice and other m modules use linear regression models. Then for any label y, Alice will achieve the oracle performance for sufficiently large rounds of assistance k in Procedure 1.

The above result is applicable to linear models and additive regression models [34] on a linear basis, e.g., spline, wavelet, or polynomial basis. Its proof is included in the supplementary material. The proof actually implies that the prediction loss decays exponentially with the rounds of assistance. The result also indicates that if the true data generating model is $\mathbb{E}(y \mid x) = \beta_a^{\mathrm{T}} x_a + \beta_b^{\mathrm{T}} x_b$, where $x = [x_a, x_b] \in \mathbb{R}^p$ with a fixed p, then Alice achieves the optimal rate $O(n^{-1})$ of prediction loss as if Alice correctly specifies the true model.

Remark 1. The results can be extended. For example, if x_a and x_b are independent, it can be proved that with one round of communications Alice can approach the oracle model with high probability for large data size n; and such an oracle loss approaches zero if $\mathbb{E}(y \mid x)$ can be written as $f_a(x_a) + f_b(x_b)$ for some functions f_a, f_b and if consistent nonparametric algorithms [35, 36] are used. Moreover, suppose that $\mathbb{E}(y \mid x)$ that cannot be written as $f_a(x_a) + f_b(x_b)$ but the interactive terms (such as $x_a \cdot x_b$ if both are scalars) involve categorical variables or continuous variables that can be well-approximated by quantizers. The Assisted Learning procedure could be modified so that Alice sends a stratified dataset to Bob which involves only additive regression functions. An illustrating example is $\mathbb{E}(y \mid x) = \beta_a x_a + \beta_b x_b + \beta_{ab} x_{a,1} x_{b,1}$ where $x_{a,1} \in \{0, 1\}$, and Alice sends data $\{x_a : x_{a,1} = 0\}$ and $\{x_a : x_{a,1} = 1\}$ separately to Bob.

4.3 Learning with Feedforward Neural Network

In this subsection, we provide an example of how feedforward neural networks can be implemented in the context of assisted learning. The data setting will be the same as described in Section 4.1. For simplicity, we consider the learning protocol of Alice and Bob with a three-layer feedforward neural network depicted in Fig. 4. Let $w_{a,k}$ (denoted by red solid lines) and $w_{b,k}$ (denoted by blue dash lines) be the input-layer weights at the kth round of assistance for Alice and Bob, respectively. Denote the rest of the weights in the neural network at the kth round of assistance by $\tilde{w}_k$.

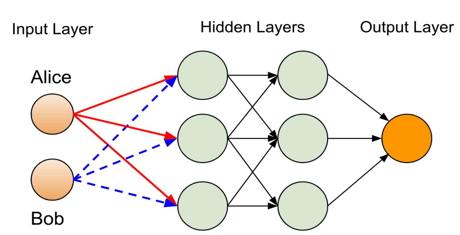


Figure 4: Feedforward neural network via assisted learning. For the weights from input to hidden layers, Alice (Bob) can only update her (his) weights denoted by red solid lines (blue dash lines).

In the *learning stage* (Stage I), at the kth round of assistance iteration, Alice calculates $w_{a,k}^{\mathrm{T}} X_A$, inquiries Bob's $w_{b,k}^{\mathrm{T}} X_B$, and then combines them to feed the neural network. If k is even, Alice will update her current weight $w_{a,k}$ (resp. $\tilde{w}_k$) to $w_{a,k+1}$ (resp. $\tilde{w}_{k+1}$). Bob will fix his weights for the next iteration, i.e. $w_{b,k+1} = w_{b,k}$. If k is odd, Alice will fix her weights for next iteration, i.e. $w_{a,k+1} = w_{a,k}$, and then sends $\tilde{w}_k$ to Bob. Bob will use the information to update his current weights $w_{b,k}$ to $w_{b,k+1}$. In the *prediction stage* (Stage II), for a new predictor x^*, Alice queries the corresponding $w_{b,K}^{\mathrm{T}} x_i^* (i \in \mathcal{S}_b)$ from Bob, and uses the trained neural network to obtain the prediction. The stop criterion we suggest is this: the above procedure of iterative assistance is repeated K times until the cross-validation error of Alice no longer decreases. The above error can be measured by, e.g., a Stage II prediction using a set of unused data (again, aligned by the data ID of Alice and Bob). Pseudocode is given in Procedure 2. Empirical evaluations on this criterion demonstrate that it typically leads to a near-optimal stopping number. The probability of choosing the optimum could be theoretically derived from large-deviation bounds under certain assumptions.

Procedure 1 Assisted learning of Module 'Alice' with m other modules (general description)

Input: Module Alice and its initial label $y \in \mathbb{R}^n$, assisting modules $\mathcal{M}_j$ for $j = 1, 2, \ldots, m$, and (optional) new predictors $\{x_t^*, t \in \mathcal{S}\}$. ($\mathcal{S}$ indexes a set of predictors, and $\mathcal{S}_j$ corresponds to module $\mathcal{M}_j$'s predictors.)
Initialisation: $e_{j,k} = y$ ($j = 1, \ldots, m$), round $k = 1$

repeat
  Alice fits a supervised model using $(e_{j,k}, X_a)$ as labeled data and model $\mathcal{A}_a$.
  Alice records its fitted model $\tilde{\mathcal{A}}_{a,j,k}$ and calculates residual $r_{j,k}$.
  for j = 1 to m **do**
    Alice sends $r_{j,k}$ to $\mathcal{M}_j$.
    $\mathcal{M}_j$ fits a supervised model using $(r_{j,k}, X_j)$ as labeled data and model $\mathcal{A}_j$.
    $\mathcal{M}_j$ records its fitted model $\tilde{\mathcal{A}}_{j,k}$ and calculates residual $\tilde{e}_{j,k}$.
    $\mathcal{M}_j$ sends $\tilde{e}_{j,k}$ to Alice.
  end for
  Alice initializes the $k+1$ round by setting $e_{j,k+1} = \tilde{e}_{j,k}$
until Stop criterion satisfied

On arrival of a new data $\{x_t^*, t \in \mathcal{S}\}$, Alice queries prediction results produced by the recorded models $\tilde{y}_k = \hat{\mathcal{A}}_{j,k}(x_t^*, t \in \mathcal{S}_j) \in \mathbb{R}^n$, for $j = 1, \ldots, m$ and $k = 1, \ldots, K$.
Alice combines (unweighted summation) the above prediction results along with its own records to form a final prediction $\tilde{y}^*$.

Output: The *Assisted Learning* prediction $\tilde{y}^*$

Procedure 2 Assisted learning of Module 'Alice' ('a') using Module 'Bob' ('b') for neural networks

Input: Module Alice, its initial label $y \in \mathbb{R}^n$, initial weights $w_{a,1}$ (of the input layer) and $\tilde{w}_1$ (of the remaining layer(s)), assisting module Bob, (optional) new predictors $\{x_t^*, t \in \mathcal{S}\}$. ($\mathcal{S}$ indexes a set of predictors, and $\mathcal{S}_a/\mathcal{S}_b$ corresponds to module Alice/Bob's predictors.)
Initialisation: round $k = 1$

repeat
  Alice calculates $w_{a,k}^{\mathrm{T}} X_A$ and receives Bob's $w_{b,k}^{\mathrm{T}} X_B$ to train the network in the following way
  if k is odd **then**
    Alice updates $w_{a,k}, \tilde{w}_k$ by using the back-propagation to obtain $w_{a,k+1}, \tilde{w}_{k+1}$, respectively
    Bob sets $w_{b,k+1} \leftarrow w_{b,k}$
  else {k is even}
    Alice sets $w_{a,k+1} \leftarrow w_{a,k}$ and sends $\tilde{w}_k$ to Bob
    Bob updates $w_{b,k}, \tilde{w}_k$ by using the back-propagation to obtain $w_{b,k+1}, \tilde{w}_{k+1}$
  end if
  Alice initializes the $k+1$ round
until Stop criterion satisfied

On arrival of a new data $\{x_t^*, t \in \mathcal{S}\}$, Alice calculates $w_{a,K}^{\mathrm{T}} x_t^* (x_t^*, t \in \mathcal{S}_a)$.
Alice queries $w_{b,K}^{\mathrm{T}} x_t^*, (x_t^*, t \in \mathcal{S}_b)$ from Bob and combine them to feed the neural network to obtain the final prediction $\tilde{y}^*$.

Output: The *Assisted Learning* prediction $\tilde{y}^*$

4.4 Relationship with Ensemble Methods

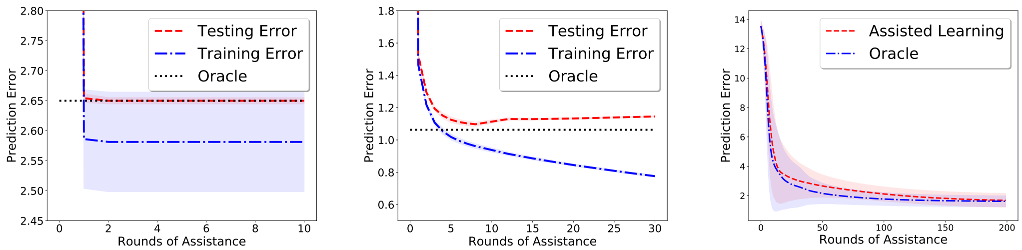
The idea of sequentially receiving assistance (e.g., residuals), building models, and combining their predictions in assisted learning is similar to ensemble machine learning techniques, e.g., Boosting [37–41] and Stacking [42–44]. In Boosting methods, each model/weak learner is built based on the same dataset only with different sample weights. In contrast, assisted learning uses side-information from heterogeneous data sources to improve a particular learner's performance.

Conceptually, stacked regressions on heterogeneous datasets highly resemble assisted learning. However, a prerequisite for stacking to work in multi-organization learning is that all participants can access labels to train the local ensemble elements. Nevertheless, in assisted learning, each participant can initiate and contribute to a task *regardless of whether it accesses labels or no*. For example, in the task of MIMIC3 (Sec. 3, Fig. 1b), the in-hospital lab aims to predict Length Of Stay. The out-side lab doesn't have access to in-hospital's private labels, but it could still initiate and assist in the in-hospital lab by fitting the received residuals instead of public labels with assisted learning protocol. The privacy-aware constraint could potentially lead to the failure of Stacked Regression in multi-organization learning. Practically, extensive experiments show that assisted learning often outperforms stacking, as summarized in Table 2.

5 Experimental Study

We provide numerical demonstrations of the proposed methods in Section 4.2 and 4.3. For synthetic data, we replicate 20 times for each method. In each replication, we trained on a dataset with size 10^3 then tested on a dataset with size 10^5. We chose a testing size much larger than the training size in order to produce a fair comparison of out-sample predictive performance [45]. For the real data, we trained on 70% of the whole data and tested on the remaining, resampled 20 times. Each line is the mean of 20 replicates, and the 'oracle score' is the testing error obtained by the model that is previously trained on the pulled data, and the shaded regions describe the corresponding ± 1 standard errors. More examples are included in the supplement.

Synthetic Data. We use the data generated by Friedman1 [46], where each data entry contains 5 features. We randomly split the features. In detail, module A has 3 features $X_A = [x_1, x_2, x_4]$, Module B has 1 feature $X_B = [x_3]$, and Module C has 1 feature $X_C = [x_5]$. In Fig. 5(a), with the least square methods, the error terms quickly converge to the oracle. In Fig. 5(b), we observe that the performance using nonlinear methods significantly improves on that in (a). In Fig. 5(c), with 2-layer neural network models, the error term of assisted learning converges slightly slower compared to the oracle, but there is negligible difference regrading the optimal prediction accuracy.



(a) Model: Linear regression (b) Model: Gradient boosting (c) Model: Two-layer NNs

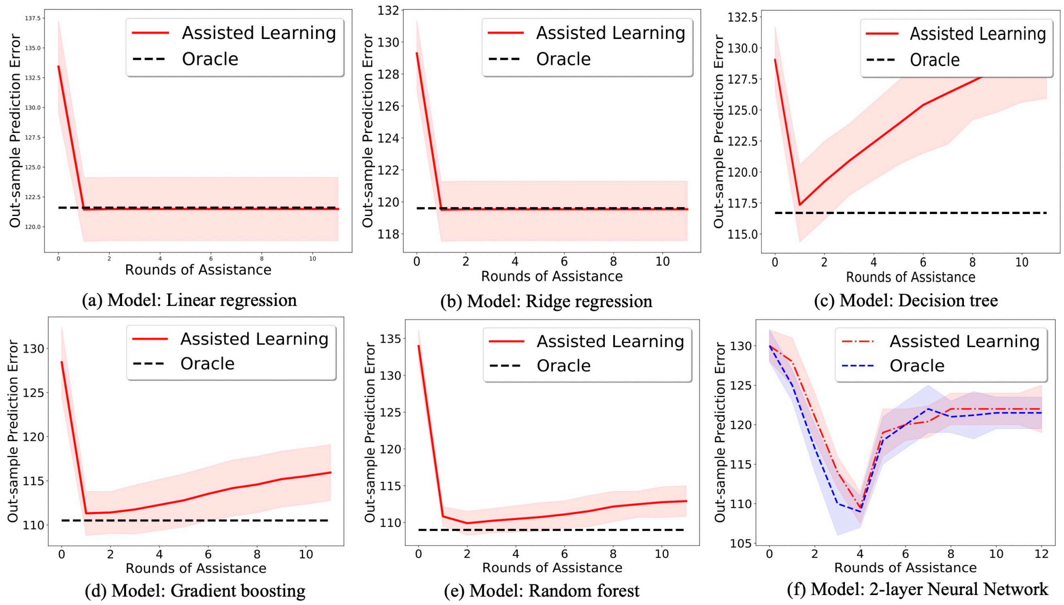
Figure 5: Out-sample prediction performance of module A via assisted learning (as measured by RMSE) against the rounds of assistance on Friedman1.

MIMIC3. We test on the task of predicting Length of Stay, one of the MIMIC3 benchmarks as described in Section 3. Following the processing procedures in [31,32], we select 17 medical features and 15000 patient cases. With Module A (Lab table) containing 3 features, module B (ICU charted event table) consisting of 12 features, and module C (Output table) holding 2 features, module B and C assist module A to predict LOS. (Features were split according to data-generating modules, namely hospital divisions). In Fig. 6(a) and 6(b), with linear/ridge regression models, the error terms converge to oracle scores in one iteration. In Fig. 6(c), 6(d), and 6(e), with decision tree and ensemble methods, the testing errors first decrease to the oracle scores and then begin to increase. Interestingly, this phenomenon resembles the classical tradeoff between underfitting and overfitting due to model complexity [12]. In our case, the rounds of assistance is the counterpart of model complexity. A solution to select an appropriate round was discussed in Section 4.2. In Fig. 6(f), with 2-layer neural network models, the error term of assisted learning converges slightly slower compared to the oracle, but there is negligible difference regrading the optimal prediction accuracy.

Comparison with the stacking method. We further test the stacked regressions on both Friedman1 and MIMIC3 in previous settings. Various sets of models are chosen for both the base model(s) and the high-level stacking models, including linear regression (LR), random forest (RF), gradient boosting (GB), neural network (NN), ridge regression (RG), and linear regression + random forest + gradient boosting (LRG). From the results summarized in Table 2, we can see that assisted learning outperform stacking methods for almost all cases.

Table 2: Prediction performances of stacking and assisted learning. Column 1 means: in assisted learning, participants use LR; in stacking, participants produce features using LR, and the meta-model combining them is LR. Columns 2-4 are similarly defined. Column 5 means: in assisted learning, each participant at each round performs model selection to choose from three models; in stacking, each participant produces three features using all the models and ensemble them using RG. Standard errors are within 0.05 over 50 replications.

Data	Friedman					MIMIC3				
Base model(s)	LR	RF	GB	GB	LRG	LR	RF	GB	GB	LRG
Model for stacking	LR	RF	GB	NN	RG	LR	RF	GB	NN	RG
Stacking	2.63	1.80	1.76	1.68	1.60	121.7	118.1	119.3	119.7	115.9
Assisted learning	2.64	1.31	1.23	1.23	1.25	120.5	109.7	111.3	111.3	110.8



(a) Model: Linear regression (b) Model: Ridge regression (c) Model: Decision tree

(d) Model: Gradient boosting (e) Model: Random forest (f) Model: 2-layer Neural Network

Figure 6: Out-sample prediction performance of module A via assisted learning (as measured by mean absolute deviation [31]) against the rounds of assistance on MIMIC3 Benchmark task predicting Length of Stay.

6 Conclusion

The interactions between multiple learners in privacy-aware scenarios pose new challenges that cannot be well addressed by classical statistical learning with a single learning objective and algorithmic procedure. In this work, we propose the notion of Assisted Learning, where the key idea is to extract task-relevant information through public labels by using private data and algorithms in order to improve learning performance and/or provide machine learning service. We include proofs and more experimental studies in the supplementary material.

Acknowledgement

The work was funded by the Army Research Office (ARO) under grant number W911NF-20-1-0222. The authors thank Hamid Krim, Mingyi Hong, and Enmao Diao for helpful discussions.

Broader Impact

The authors envision the following positive ethical and societal consequences. First, the developed concepts, methods, and theories will potentially benefit fields such as engineering and epidemiology that often involve multi-organizational collaborations, since they may not need to share their private data. Second, the work will also benefit the general public whose private data are often held by various organizations. The authors cannot think of a negative ethical or societal consequence of this work.

References

[1] J. T. Farrar, A. B. Troxel, K. Haynes, I. Gilron, R. D. Kerns, N. P. Katz, B. A. Rappaport, M. C. Rowbotham, A. M. Tierney, D. C. Turk *et al.*, "Effect of variability in the 7-day baseline pain diary on the assay sensitivity of neuropathic pain randomized clinical trials: an acttion study," *Pain®*, vol. 155, no. 8, pp. 1622–1631, 2014.

[2] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[3] B. Lo, "Sharing clinical trial data: maximizing benefits, minimizing risk," *Jama*, vol. 313, no. 8, pp. 793–794, 2015.

[4] D. Dahl, G. G. Wojtal, M. J. Breslow, D. Huguez, D. Stone, and G. Korpi, "The high cost of low-acuity icu outliers," *J. Healthc. Manag.*, vol. 57, no. 6, pp. 421–433, 2012.

[5] M. M. Alabbadi, "Mobile learning (mlearning) based on cloud computing: mlearning as a service (mlaas)," in *Proc. UBICOMM*, 2011.

[6] M. Ribeiro, K. Grolinger, and M. A. Capretz, "Mlaas: Machine learning as a service," in *Proc. ICMLA*. IEEE, 2015, pp. 896–902.

[7] C. Dwork and K. Nissim, "Privacy-preserving datamining on vertically partitioned databases," in *CRYPTO*. Springer, 2004, pp. 528–544.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*. Springer, 2006, pp. 486–503.

[9] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[10] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[11] Y. Yang, "Regression with multiple candidate models: selecting or mixing?" *Statistica Sinica*, pp. 783–809, 2003.

[12] J. Ding, V. Tarokh, and Y. Yang, "Model selection techniques–an overview," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 16–34, 2018.

[13] J. Vaidya and C. Clifton, "Privacy preserving naive bayes classifier for vertically partitioned data," in *Proc. ICDM*. SIAM, 2004, pp. 522–526.

[14] J. Vaidya, "A survey of privacy-preserving methods across vertically partitioned data," in *Privacy-preserving data mining*. Springer, 2008, pp. 337–358.

[15] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. CCS*. ACM, 2015, pp. 1310–1321.

[16] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[17] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[18] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "Secureboost: A lossless federated learning framework," *arXiv preprint arXiv:1901.08755*, 2019.

[19] Z. Fengy, H. Xiong, C. Song, S. Yang, B. Zhao, L. Wang, Z. Chen, S. Yang, L. Liu, and J. Huan, "Securegbm: Secure multi-party gradient boosting," *arXiv preprint arXiv:1911.11997*, 2019.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *Proc. TIST*, vol. 10, no. 2, p. 12, 2019.

[21] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient vertical federated learning framework," *arXiv preprint arXiv:1912.11187*, 2019.

[22] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. SFCS*, 1986, pp. 162–167.

[23] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, 1998.

[24] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Secure linear regression on vertically partitioned datasets." *IACR Cryptology ePrint Archive*, vol. 2016, p. 892, 2016.

[25] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proc. IEEE SP*. IEEE, 2017, pp. 19–38.

[26] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Privacy-preserving distributed linear regression on high-dimensional data," *Proc. PET*, vol. 2017, no. 4, pp. 345–364, 2017.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," 2011.

[28] T. Baltrusaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[29] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[30] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[31] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*, 2017.

[32] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Inform.*, vol. 83, pp. 112–134, 2018.

[33] G. Claeskens, N. L. Hjort *et al.*, "Model selection and model averaging," *Cambridge Books*, 2008.

[34] C. J. Stone, "Additive regression and other nonparametric models," *Ann. Stat.*, pp. 689–705, 1985.

[35] ——, "Consistent nonparametric regression," *Ann. Stat.*, pp. 595–620, 1977.

[36] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, no. Apr, pp. 1063–1095, 2012.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[38] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[39] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[40] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[41] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[42] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[43] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[44] M. LeBlanc and R. Tibshirani, "Combining estimates in regression and classification," *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1641–1650, 1996.

[45] J. Ding, V. Tarokh, and Y. Yang, "Model selection techniques: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 16–34, 2018.

[46] J. H. Friedman *et al.*, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.