Privacy-preserving Stacking with Application to Cross-organizational Diabetes Prediction

Quanming Yao[1,2], **Xiawei Guo**[1], **James Kwok**[2], **Weiwei Tu**[1], **Yuqiang Chen**[1], **Wenyuan Dai**[1] and **Qiang Yang**[2]

[1]4Paradigm Inc

[2]Department of Computer Science and Engineering, HKUST

Abstract

To meet the standard of differential privacy, noise is usually added into the original data, which inevitably deteriorates the predicting performance of subsequent learning algorithms. In this paper, motivated by the success of improving predicting performance by ensemble learning, we propose to enhance privacy-preserving logistic regression by stacking. We show that this can be done either by sample-based or feature-based partitioning. However, we prove that when privacy-budgets are the same, feature-based partitioning requires fewer samples than sample-based one, and thus likely has better empirical performance. As transfer learning is difficult to be integrated with a differential privacy guarantee, we further combine the proposed method with hypothesis transfer learning to address the problem of learning across different organizations. Finally, we not only demonstrate the effectiveness of our method on two benchmark data sets, i.e., MNIST and NEWS20, but also apply it into a real application of cross-organizational diabetes prediction from RUIJIN data set, where privacy is of a significant concern. [1]

1 Introduction

In recent years, data privacy has become a serious concern in both academia and industry [Dwork *et al.*, 2006; Chaudhuri *et al.*, 2011; Dwork and Roth, 2014; Abadi *et al.*, 2016]. There are now privacy laws, such as Europe's General Data Protection Regulation (GDPR), which regulates the protection of private data and restricts data transmission between organizations. These raise challenges for cross-organizational machine learning [Pathak *et al.*, 2010; Hamm *et al.*, 2016; Papernot *et al.*, 2017; Xie *et al.*, 2017], in which data have to be distributed to different organizations, and the learning model needs to make predictions in private.

A number of approaches have been proposed to ensure privacy protection. In machine learning, differential privacy [Dwork and Roth, 2014] is often used to allow data be exchanged among organizations. To design a differentially private algorithm, carefully designed noise is usually added to the original data to disambiguate the algorithms. Many standard learning algorithms have been extended for differential privacy. These include logistic regression [Chaudhuri *et al.*, 2011], trees [Emekçi *et al.*, 2007; Fong and Weber-Jahnke, 2012], and deep networks [Shokri and Shmatikov, 2015; Abadi *et al.*, 2016]. In particular, linear models are simple and easy to understand, and their differentially private variants (such as privacy-preserving logistic regression (PLR)) [Chaudhuri *et al.*, 2011]) have rigorous theoretical guarantees [Chaudhuri *et al.*, 2011; Bassily *et al.*, 2014; Hamm *et al.*, 2016; Kasiviswanathan and Jin, 2016]. However, the injection of noise often degrades prediction performance.

Ensemble learning can often signficantly improve the performance of a single learning model [Zhou, 2012]. Popular examples include bagging [Breiman, 1996a], boosting [Friedman *et al.*, 2000], and stacking [Wolpert, 1992]. These motivate us to develop an ensemble-based method which can benefit from data protection, while enjoying good prediction performance. Bagging and boosting are based on partitioning of training samples, and use pre-defined rules (majority or weighted voting) to combine predictions from models trained on different partitions. Bagging improves learning performance by reducing the variance. Boosting, on the other hand, is useful in converting weak models to a strong one. However, the logistic regression model, which is the focus in this paper, often has good performance in many applications, and is a relatively strong classifier. Besides, it is a convex model and relatively stable.

Thus, in this paper, we focus on stacking. While stacking also partitions the training data, this can be based on either samples [Breiman, 1996b; Smyth and Wolpert, 1999; Ozay and Vural, 2012] or features [Boyd *et al.*, 2011]. Multiple low-level models are then learned on the different data partitions, and a high-level model (typically, a logistic regression model) is used to combine their predictions. By combining with PLR, we show how differential privacy can be ensured in stacking. Besides, when the importance of features is known a priori, they can be easily incorporated in feature-based partitioning. We further analyze the learning guarantee of sample-based and feature-based stacking, and show theoretically that feature-based partitioning can have lower sample complexity (than sample-based partitioning), and thus better performance. By adapting the feature impor-

[1]Correspondence to X. Guo at guoxiawei@4paradigm.com

tance, its learning performance can be further boosted.

To demonstrate the superiority of the proposed method, we perform experiments on two benchmark data sets (MNIST and NEWS20). Empirical results confirm that feature-based stacking performs better than sample-based stacking. It is also better than directly using PLR on the training data set. Besides, the prediction performance is further boosted when feature importance is used. Finally, we apply the proposed approach for cross-organizational diabetes prediction in the transfer learning setting. The experiment is performed on the RUIJIN data set, which contains over ten thousands diabetes records from across China. Results show significantly improved diabetes prediction performance over the state-of-the-art, while still protecting data privacy.

Notation. In the sequel, vectors are denoted by lowercase boldface, and $(\cdot)^\top$ denotes transpose of a vector/matrix; $\sigma(a) = {\exp(a)}/{(1+\exp(a))}$ is the sigmoid function. A function g is μ-strongly convex if $g(\alpha\mathbf{w} + (1-\alpha)\mathbf{u}) \le \alpha g(\mathbf{w}) + (1-\alpha)g(\mathbf{u}) - \frac{\mu}{2}\alpha(1-\alpha)\|\mathbf{w}-\mathbf{u}\|^2$ for any $\alpha \in (0,1)$.

2 Related Works

2.1 Differential Privacy

Differential privacy [Dwork *et al.*, 2006; Dwork and Roth, 2014] has been established as a rigorous standard to guarantee privacy for algorithms that access private data. Intuitively, given a privacy budget ϵ, an algorithm preserves ϵ-differentially privacy if changing one entry in the data set does not change the likelihood of any of the algorithm's output by more than ϵ. Formally, it is defined as follows.

Definition 1 ([Dwork *et al.*, 2006])**.** *A randomized mechanism M is ϵ-differentially private if for all output t of M and for all input data $\mathcal{D}_1, \mathcal{D}_2$ differing by one element,* $\Pr(M(\mathcal{D}_1) = t) \le e^\epsilon \Pr(M(\mathcal{D}_2) = t)$.

To meet the ϵ-differentially privacy guarantee, careful perturbation or noise usually needs to be added to the learning algorithm. A smaller ϵ provides stricter privacy guarantee but at the expense of heavier noise, leading to larger performance deterioration [Chaudhuri *et al.*, 2011; Bassily *et al.*, 2014]. A relaxed version of ϵ-differentially private, called (ϵ,δ)-differentially privacy in which δ measures the loss in privacy, is proposed [Dwork and Roth, 2014]. However, we focus on the more stringent Definition 1 in this paper.

2.2 Privacy-preserving Logistic Regression (PLR)

Logistic regression has been popularly used in machine learning [Friedman *et al.*, 2012]. Various differential privacy approaches have been developed for logistic regression. Examples include output perturbation [Dwork *et al.*, 2006; Chaudhuri *et al.*, 2011], gradient perturbation [Abadi *et al.*, 2016] and objective perturbation [Chaudhuri *et al.*, 2011; Bassily *et al.*, 2014]. In particular, objective perturbation, which adds designed and random noise to the learning objective, has both privacy and learning guarantees as well as good empirical performance.

Privacy-preserving logistic regression (PLR) [Chaudhuri *et al.*, 2011] is the state-of-the-art model based on objective perturbation. Given a data set $\mathcal{D} = \{\mathbf{x}_i,\ y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the sample and y_i the corresponding class label, we first consider the regularized risk minimization problem:

$$\min_{\mathbf{w}} {1}/{n} \sum\nolimits_{i=1}^n \ell(\mathbf{w}^\top\mathbf{x}_i, y_i) + \lambda g(\mathbf{w}), \tag{1}$$

where $\mathbf{w}$ is a vector of the model parameter, $\ell(\hat{y}, y) = \log(1+e^{-y\hat{y}})$ is the logistic loss (with predicted label $\hat{y}$ and given label y), g is the regularizer and $\lambda \ge 0$ is a hyperparameter. To guarantee privacy, Chaudhuri *et al.* (2011) added two extra terms to (1), leading to:

$$\min_{\mathbf{w}} {1}/{n} \sum_{i=1}^n \ell(\mathbf{w}^\top\mathbf{x}_i, y_i) + {\mathbf{b}^\top\mathbf{w}}/{n} + {\Delta\|\mathbf{w}\|^2}/{2} + \lambda g(\mathbf{w}), \tag{2}$$

where $\mathbf{b}$ is random noise drawn from $h(\mathbf{b}) \propto \exp({\epsilon'}/{2}\|\mathbf{b}\|)$ with $\mathbb{E}(\|\mathbf{b}\|) = {2d}/{\epsilon'}$, ϵ' is a privacy budget modified from ϵ, and Δ is a scalar depending on λ, n, ϵ. The whole PLR procedure is shown in Algorithm 1.

Algorithm 1 PLR: Privacy-preserving logistic regression.

Require: privacy budget ϵ, data set $\mathcal{D}$;
$\epsilon' = \epsilon - \log(1 + {1}/{2n\lambda} + {1}/{16n^2\lambda^2})$;
if $\epsilon' > 0$ **then**
  $\Delta = 0$;
else
  $\Delta = (4n(\exp({\epsilon}/{4}) - 1))^{-1} - \lambda$ and $\epsilon' = {\epsilon}/{2}$;
end if
scale $\|\mathbf{x}\| \le 1$ for all $\mathbf{x} \in \mathcal{D}$;
pick a random vector $\mathbf{b}$ from $h(\mathbf{b}) \propto \exp\left({\epsilon'\|\mathbf{b}\|}/{2}\right)$;
obtain $\mathbf{w}$ by solving (2);
return $\mathbf{w}$.

Proposition 1 ([Chaudhuri *et al.*, 2011])**.** *If the regularizer g is strongly convex, Algorithm 1 provides ϵ-differential privacy.*

While privacy guarantee is desirable, the resultant privacy-preserving machine learning model may not have good learning performance. In practice, the performance typically degrades dramatically because of the introduction of noise [Chaudhuri *et al.*, 2011; Rajkumar and Agarwal, 2012; Bassily *et al.*, 2014; Shokri and Shmatikov, 2015]. Assume that samples from $\mathcal{D}$ are drawn i.i.d. from an underlying distribution P. Let $L(\mathbf{w}; P) = \mathbb{E}_{(\mathbf{x},y)\sim P}[\ell(\mathbf{w}^\top\mathbf{x}, y)]$ be the expected loss of the model. The following Proposition shows the number of samples needed for PLR to have comparable performance as a given baseline model.

Proposition 2 ([Chaudhuri *et al.*, 2011])**.** *Let $g(\cdot) = {1}/{2}\|\cdot\|^2$, and $\mathbf{v}$ be a reference model parameter. Given $\delta > 0$ and $\epsilon_g > 0$, there exists a constant C_1 such that when*

$$n > C_1 \max\left({\|\mathbf{v}\|^2 \log(\frac{1}{\delta})}/{\epsilon_g^2},\ {d\log(\frac{d}{\delta})\|\mathbf{v}\|}/{\epsilon_g\epsilon},\ {\|\mathbf{v}\|^2}/{\epsilon_g\epsilon} \right), \tag{3}$$

$\mathbf{w}$ *from Algorithm 1 meets* $\Pr[L(\mathbf{w},P) \le L(\mathbf{v},P) + \epsilon_g] \ge 1-\delta$.

2.3 Multi-Party Data Learning

Ensemble learning has been considered with differential privacy under multi-party data learning (MPL). The task is to combine predictors from multiple parties with privacy [Pathak *et al.*, 2010]. Pathak *et al.* [2010] first proposed a specially designed protocol to privately combine multiple predictions. The performance is later surpassed by [Hamm *et al.*, 2016; Papernot *et al.*, 2017], which uses another classifier built on auxiliary unlabeled data. However, all these combination methods rely on extra, privacy-insensitive public data, which may not be always available. Moreover, the aggregated prediction may not be better than the best single party's prediction. There are also MPL methods that do not use ensemble learning. Rajkumar and Agarwal [2012] used stochastic gradient descent, and Xie *et al.* [2017] proposed a multi-task learning method. While these improve the performance of the previous ones based on aggregation, they gradually lose the privacy guarantee after more and more iterations.

3 Privacy-preserving Ensemble

In this section, we propose to improve the learning guarantee of PLR by ensemble learning [Zhou, 2012]. Popular examples include bagging [Breiman, 1996a], boosting [Friedman *et al.*, 2000], and stacking [Wolpert, 1992]. Bagging and boosting are based on partitioning of training samples, and use pre-defined rules (majority or weighted voting) to combine predictions from models trained on different partitions. Bagging improves learning performance by reducing the variance. However, logistic regression is a convex model and relatively stable. Boosting, on the other hand, is useful in combining weak models to a strong one, while logistic regression is a relatively strong classifier and often has good performance in many applications.

3.1 Privacy-preserving Stacking with Sample Partitioning (SP)

We first consider using stacking with SP, and PLR is used as both the low-level and high-level models (Algorithm 2). As stacking does not impose restriction on the usage of classifiers on each partition of the training data, a simple combination of stacking and PLR can be used to provide privacy guarantee.

Algorithm 2 PST-S: Privacy-preserving stacking with SP.

Require: privacy budget ϵ, data set $\mathcal{D}$;
1: partition $\mathcal{D}$ into disjoint sets $\mathcal{D}^l$ and $\mathcal{D}^h$, for training of the low-level and high-level models, respectively;
2: partition samples in $\mathcal{D}^l$ to K disjoint sets $\{\mathcal{S}_1, \dots, \mathcal{S}_K\}$;
3: **for** $k = 1, \dots, K$ **do**
4: train PLR (Algorithm 1) with privacy budget ϵ on $\mathcal{S}_k$, and obtain the low-level model parameter $\mathbf{w}_k^l$;
5: **end for**
6: construct meta-data set $\mathcal{M}^s = \{[\sigma(\mathbf{x}^\top\mathbf{w}_1^l);\dots;\sigma(\mathbf{x}^\top\mathbf{w}_K^l)], y\}$ using all samples $\{\mathbf{x}, y\} \in \mathcal{D}^h$;
7: train PLR (Algorithm 1) with privacy budget ϵ on $\mathcal{M}^s$, and obtain the high-level model parameter $\mathbf{w}^h$;
8: **return** $\{\mathbf{w}_k^l\}$ and $\mathbf{w}^h$.

Proposition 3. *If the regularizer g is strongly convex, Algorithm 2 provides ϵ-differential privacy.*

However, while the high-level model can be better than any of the single low-level models [Džeroski and Ženko, 2004], Algorithm 2 may not perform better than directly using PLR on the whole $\mathcal{D}$ for the following two reasons. First, each low-level model uses only $\mathcal{S}_k$ (step 4), which is about $1/K$ the size of $\mathcal{D}$ (assuming that the data set $\mathcal{D}$ is partitioned uniformly). This smaller sample size may not satisfy condition (3) in Proposition 2. Second, in many real-world applications, features are not of equal importance. For example, for diabetes prediction using the RUIJIN data set (Table 3), Glu120 and Glu0, which directly measure glucose levels in the blood, are more relevant than features such as age and number of children. However, during training of the low-level models, Algorithm 2 adds equal amounts of noise to all features. If we can add less noise to the more important features while keeping the same privacy guarantee, we are likely to get better learning performance.

3.2 Privacy-preserving Stacking with Feature Partitioning (FP)

To address the above problems, we propose to partition the data based on features instead of samples in training the low-level models. The proposed feature-based stacking approach is shown in Algorithm 3. Features are partitioned into K subsets, and $\mathcal{D}^l$ is split correspondingly into K disjoint sets $\{\mathcal{F}_1, \dots, \mathcal{F}_K\}$. Obviously, as the number of training samples is not reduced, the sample size condition for learning performance guarantee is easier to be satisfied (details will be established in Theorem 4).

Algorithm 3 PST-F: Privacy-preserving stacking with FP.

Require: privacy budget ϵ, data set $\mathcal{D}$, feature importance $\{q_k\}_{k=1}^K$ where $q_k \geq 0$ and $\sum_{k=1}^K q_k = 1$; [2]
1: partition $\mathcal{D}$ into disjoint sets $\mathcal{D}^l$ and $\mathcal{D}^h$, for training of the low-level model and high-level model, respectively;
2: partition $\mathcal{D}^l$ to K disjoint sets $\{\mathcal{F}_1, \dots, \mathcal{F}_K\}$ based on features;
3: $\epsilon' = \epsilon - \sum_{k=1}^K \log(1 + q_k^2/2n\lambda_k + q_k^4/16n^2\lambda_k^2)$;
4: **for** $k = 1, \dots, K$ **do**
5: scale $\|\mathbf{x}\| \leq q_k$ for all $\mathbf{x} \in \mathcal{F}_k$;
6: **if** $\epsilon' > 0$ **then**
7: $\Delta_k = 0$ and $\epsilon_k = \epsilon'$;
8: **else**
9: $\Delta_k = q_k^2/4n(\exp(\epsilon q_k/4)-1) - \lambda_k$ and $\epsilon_k = \epsilon/2$;
10: **end if**
11: pick a random $\mathbf{b}_k$ from $h(\mathbf{b}) \propto \exp(\epsilon_k\|\mathbf{b}\|/2)$;
12: $\mathbf{w}_k^l = \arg\min_{\mathbf{w}} 1/n \sum_{\mathbf{x}_i \in \mathcal{F}_k} \ell(\mathbf{w}^\top\mathbf{x}_i, y_i) + \mathbf{b}_k^\top\mathbf{w}/n + \Delta\|\mathbf{w}\|^2/2 + \lambda_k g_k(\mathbf{w})$;
13: **end for**
14: construct meta-data set $\mathcal{M}^f = \{[\sigma(\mathbf{x}_{(1)}^\top\mathbf{w}_1^l),\dots,\sigma(\mathbf{x}_{(K)}^\top\mathbf{w}_K^l)], y\}$ using all $\{\mathbf{x}, y\} \in \mathcal{D}^h$, where $\mathbf{x}_{(k)}$ is a vector made from $\mathbf{x}$ by taking features covered by $\mathcal{F}_k$;
15: train PLR (Algorithm 1) with privacy budget ϵ on $\mathcal{M}^f$, and obtain the high-level model parameter $\mathbf{w}^h$;
16: **return** $\{\mathbf{w}_k^l\}$ and $\mathbf{w}^h$.

When the relative importance of feature subsets is known, Algorithm 3 adds less noise to the more important features. Specifically, let the importance[3] of $\mathcal{F}_k$ (with d_k features) be q_k, where $q_k \geq 0$ and $\sum_{k=1} q_k = 1$, and is independent with $\mathcal{D}$. Assume that $\epsilon' > 0$ in step 6 (and thus $\epsilon_k = \epsilon'$). Recall

[3] When feature importance is not known, $q_1 = \cdots = q_K = 1/K$.

from Section 2.2 that $\mathbb{E}(\|\mathbf{b}_k\|) = {}^{2d_k}/_{\epsilon_k} = {}^{2d_k}/_{\epsilon'}$. By scaling the samples in each $\mathcal{F}_k$ as in step 5, the injected noise level in $\mathcal{F}_k$ is given by ${}^{\mathbb{E}(\|\mathbf{b}_k\|)}/_{\|\mathbf{x}\|} = {}^{2d_k}/_{\epsilon' q_k}$. This is thus inversely proportional to the importance q_k.

Remark 1. *In the special case where only one feature group has nonzero importance, Algorithm 3 reduces Algorithm 1 on that group, and privacy is still guaranteed.*

Finally, a privacy-preserving low-level logistic regression model is obtained in step 12, and a privacy-preserving high-level logistic regression model is obtained in step 15. Theorem 4 guarantees privacy of Algorithm 3. Note that the proofs in [Chaudhuri *et al.*, 2011; Bassily *et al.*, 2014] cannot be directly used, as they consider neither stacking nor feature importance.

Theorem 4. *If all g_k's are strongly convex, Algorithm 3 provides ϵ-differential privacy.*

Analogous to Proposition 1, the following bounds the learning performance of each low-level model.

Theorem 5. $g_k = {}^1/_2\| \cdot - \mathbf{u}_k\|^2$, *where $\mathbf{u}_k$ is any constant vector, and $\mathbf{v}_k$ is a reference model parameter. Let $a_k = q_k\|\mathbf{v}_k\|$. given $\delta > 0$ and $\epsilon_g > 0$, there exists a constant C_1 such that when*

$$n > C_1 \max\left({}^{a_k^2 \log(1/\delta)}/_{\epsilon_g^2}, {}^{d \log(d/K\delta) a_k}/_{q_k K \epsilon_g \epsilon}, {}^{a_k^2}/_{\epsilon_g \epsilon} \right), \quad (4)$$

$\mathbf{w}_k^l$ *from Algorithm 3 satisfies* $\Pr[L(\mathbf{w}_k^l, P) \leq L(\mathbf{v}_k, P) + \epsilon_g] \geq 1 - \delta$.

Remark 2. *When $K = 1$ (a single low-level model trained with all features) and $\mathbf{u}_k = \mathbf{0}$, Theorem 5 reduces to Proposition 2.*

Note that, to keep the same bound $L(\mathbf{v}_k, P) + \epsilon_g$, since $\mathbf{x}$s' are scaled by q_k, $\mathbf{v}_k$ should be scaled by ${}^1/_{q_k}$, so $\mathbb{E}(a_k) = \mathbb{E}(q_k\|\mathbf{v}_k\|)$ remains the same as q_k changes. Thus, Theorem 5 shows that low-level models on more important features can indeed learn better, if these features are assigned with larger q_k. Since stacking can have better performance than any single model [Ting and Witten, 1999; Džeroski and Ženko, 2004] and Theorem 5 can offer better learning guarantee than Proposition 2, Algorithm 3 can have better performance than Algorithm 1. Finally, compared with Proposition 1, g_k in theorem 5 is more flexible in allowing an extra $\mathbf{u}_k$. We will show in Section 3.3 that this is useful for transfer learning.

Since the learning performance of stacking itself is still an open issue [Ting and Witten, 1999], we leave the guarantee for the whole Algorithm 3 as future work. A potential problem with FP is that possible correlations among feature subsets can no longer be utilized. However, as the high-level model can combine information from various low-level models, empirical results in Section 4.1 show that this is not problematic unless K is very large.

3.3 Application to Transfer Learning

Transfer learning [Pan and Yang, 2010] is a powerful and promising method to extract useful knowledge from a source domain to a target domain. A popular transfer learning approach is hypothesis transfer learning (HTL) [Kuzborskij and Orabona, 2013], which encourages the hypothesis learned in the target domain to be similar with that in the source domain. For application to (1), HTL adds an extra regularizer as:

$$\min_{\mathbf{w}} \sum\nolimits_{\mathbf{x}_i \in \mathcal{D}_{\text{tgt}}} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \lambda g(\mathbf{w}) + {}^{\eta}/_{2}\|\mathbf{w} - \mathbf{w}_{\text{src}}\|^2. \quad (5)$$

Here, η is a hyperparameter, $\mathcal{D}_{\text{tgt}}$ is the target domain data, and $\mathbf{w}_{\text{src}}$ is obtained from the source domain. Algorithm 4 shows how PST-F can be extended with HTL using privacy budgets ϵ_{src} and ϵ_{tgt} for the source and target domains, respectively. The same feature partitioning is used on both the source and target data. PLR is trained on each source domain data subset to obtain $(\mathbf{w}_{\text{src}})_k$ (steps 2-4). This is then transferred to the target domain using PST-F with $g_k(\mathbf{w}) = \frac{1}{2}\|\mathbf{w} - (\mathbf{w}_{\text{src}})_k\|^2$ (step 5).

Algorithm 4 PST-H: Privacy-preserving stacking with HTL.

Require: source data sets $\mathcal{D}_{\text{src}}$, target data set $\mathcal{D}_{\text{tgt}}$, and corresponding privacy budgets ϵ_{src} and ϵ_{tgt}, respectively.
(source domain processing)
1: partition $\mathcal{D}_{\text{src}}$ to K disjoint sets $\{\mathcal{F}_1, \dots, \mathcal{F}_K\}$ based on features;
2: **for** $k = 1, \dots, K$ **do**
3: train PLR with privacy budget ϵ_{src} on $\mathcal{F}_k$ and obtain $(\mathbf{w}_{\text{src}})_k$;
4: **end for**
(target domain processing)
5: obtain $\{(\mathbf{w}_{\text{tgt}})_k^l\}$ and $\mathbf{w}_{\text{tgt}}^h$ from PST-F (Algorithm 3) by taking $g_k(\mathbf{w}) = {}^1/_2\|\mathbf{w} - (\mathbf{w}_{\text{src}})_k\|^2$ and privacy budget ϵ_{tgt} on $\mathcal{D}_{\text{tgt}}$;
6: **return** $\{(\mathbf{w}_{\text{src}})_k\}$ for source domain, $\{(\mathbf{w}_{\text{tgt}})_k^l\}$ and $\mathbf{w}_{\text{tgt}}^h$ for target domain.

The following provides privacy guarantees on both the source and target domains. Recently, privacy-preserving HTL is also proposed in [Wang *et al.*, 2018]. However, it does not consider stacking and ignores feature importance.

Corollary 6. *Algorithm 4 provides ϵ_{src}- and ϵ_{tgt}-differential privacy guarantees for the source and target domains.*

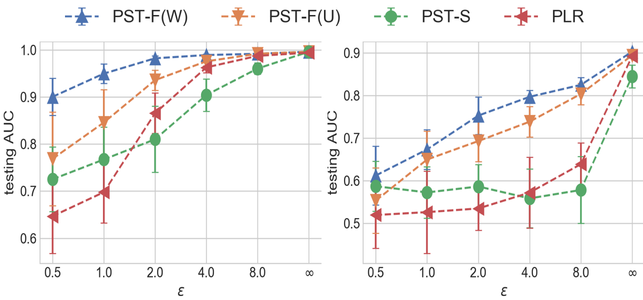
4 Experiments

4.1 Benchmark Datasets

Experiments are performed on two popular benchmark data sets for evaluating privacy-preserving learning algorithms [Shokri and Shmatikov, 2015; Papernot *et al.*, 2017; Wang *et al.*, 2018]: MNIST [LeCun *et al.*, 1998] and NEWS20 [Lang, 1995] (Table 1). The MNIST data set contains images of handwritten digits. Here, we use the digits 0 and 8. We randomly select 5000 samples. 60% of them are used for training (with ${}^1/_3$ of this used for validation), and the remaining 20% for testing. The NEWS20 data set is a collection of newsgroup documents. Documents belonging to the topic "sci" are taken as positive samples, while those in the topic "talk" are taken as negative. Finally, we use PCA to reduce the feature dimensionality to 100, as original dimensionality for MINIST/NEWS20 is too high for differentially private algorithms to handle as the noise will be extremely large. Note that we use PCA for simplicity of the ablation study. However, note that the importance scores should be obtained from side information independent from

MNIST			NEWS20		
#train	#test	#features	#train	#test	#features
3000	2000	100	4321	643	100

Table 1: Summary of the MNIST and NEWS20 data sets.



(a) MNIST. (b) NEWS20.

Figure 1: Testing AUC vs ϵ. Here, "∞" corresponds to the non-privacy-preserving version of the corresponding algorithms.

the data or from experts' opinions (as in diabetes example). Otherwise, ϵ-differential privacy will not be guaranteed.

The following algorithms are compared: (i) PLR, which applies Algorithm 1 on the training data; (ii) PST-S: Algorithm 2, based on SP; and (iii) PST-F: Algorithm 3, based on FP. We use $K = 5$ and 50% of the data for $\mathcal{D}^l$ and the remaining for $\mathcal{D}^h$. Two PST-F variants are compared: PST-F(U), with random FP and equal feature importance. And PST-F(W), with partitioning based on the PCA feature scores; and the importance of the kth group $\mathcal{F}_k$ is

$$q_k = \sum_{i:f_i \in \mathcal{F}_k} v_i / \sum_{j:f_j \in \mathcal{D}^l} v_j, \quad (6)$$

where v_i is the variance of the ith feature f_i. Gradient perturbation is worse than objective perturbation in logistic regression [Bassily *et al.*, 2014], thus is not compared.

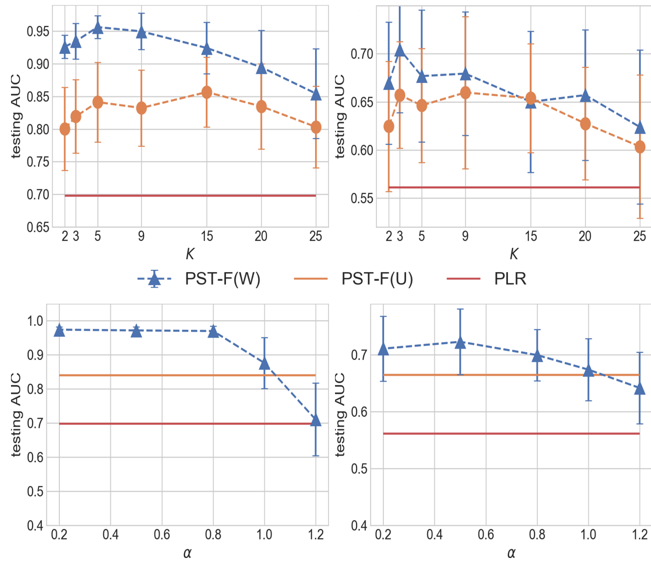
The area-under-the-ROC-curve (AUC) [Hanley and McNeil, 1983] on the testing set is used for performance evaluation. Hyper-parameters are tuned using the validation set. To reduce statistical variations, the experiment is repeated 10 times, and the results averaged.

Varying Privacy Budget ϵ

Figure 1 shows the testing AUC's when the privacy budget ϵ is varied. As can be seen, the AUCs for all methods improve when the privacy requirement is relaxed (ϵ is large and less noise is added). Moreover, PST-S can be inferior to PLR, due to insufficient training samples caused by SP. Both PST-F(W) and PST-F(U) have better AUCs than PST-S and PLR. In particular, PST-F(W) is the best as it can utilize feature importance. Since PST-S is inferior to PST-F(U), we only consider PST-F(U) in the following experiments.

Varying Number of Partitions K

In this experiment, we fix $\epsilon = 1$, and vary K. As can be seen from Figure 2(a)-(b), when K is very small, ensemble learning is not effective. When K is too large, a lot of feature correlation information is lost and the testing AUC also decreases.



(a) MNIST. (b) NEWS20.

Figure 2: Testing AUC with different K (first row) and different feature importance settings (second row).

Changing the Feature Importance

In the above experiments, feature importance is defined based on the variance from PCA. Here, we show how feature importance influences prediction performance. In real-world applications, we may not know the exact importance of features. Thus, we replace variance v_i by the ith power of α (α^i), where α is a positive constant, and use (6) for assigning weights. Note that when $\alpha < 1$, more importance features have larger weights; and vice versa when $\alpha > 1$. Note that PST-F(W) does not reduce to PST-F(U) when $\alpha = 1$, as more important features are still grouped together. Figure 2(c)-(d) show the testing AUCs at different α's. As can be seen, with proper assigned weights (i.e., $\alpha < 1$ and more important features have larger q_k's), the testing AUC can get higher. If less important features are more valued, the testing AUC decreases and may not be better than PST-F(U), which uses uniform weights. Moreover, we see that PST-F(W) is not sensitive to the weights once they are properly assigned.

Choice of High-Level Model

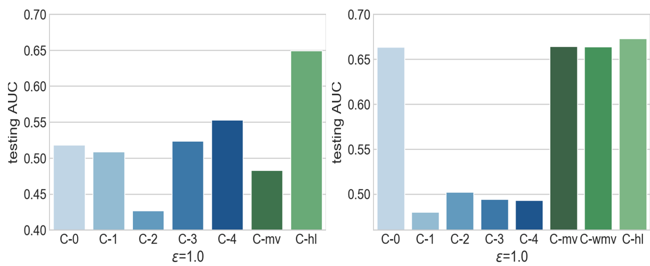
We compare different high-level models in combining predictions from the low-level models. The following methods are compared: (i) major voting (C-mv) from low-level models; (ii) weighted major voting (C-wmv), which uses $\{q_k\}$ as the weights; and (iii) by a high-level model in PST-F (denoted "C-hl"). Figure 3 shows results on NEWS20 with $\epsilon = 1.0$. As can be seen, C-0 in Figure 3(b) has the best performance among all single low-level models, as it contains the most important features. Besides, stacking (i.e., C-hl), is the best way to combine predictions from C-{0-4}, which also offers better performance than any single low-level models.

4.2 Diabetes Prediction

Diabetes is a group of metabolic disorders with high blood sugar levels over a prolonged period. The RUIJIN diabetes

branch#	1	2	3	4	5	6	7	8
PST-H(W)	**0.747±0.032**	**0.736±0.032**	**0.740±0.040**	**0.714±0.040**	**0.766±0.039**	**0.707±0.017**	**0.721±0.0464**	**0.753±0.042**
PST-H(U)	0.678±0.049	**0.724±0.037**	0.652±0.103	**0.708±0.033**	0.653±0.070	0.663±0.036	**0.682±0.0336**	0.692±0.044
PPHTL	0.602±0.085	0.608±0.078	0.528±0.062	0.563±0.067	0.577±0.075	0.601±0.031	0.580±0.0708	0.583±0.056
PLR(target)	0.548±0.088	0.620±0.055	0.636±0.046	0.579±0.075	0.533±0.058	0.613±0.035	0.561±0.0764	0.584±0.045
branch#	9	10	11	12	13	14	15	16
PST-H(W)	**0.701±0.023**	**0.698±0.036**	**0.736±0.046**	**0.738±0.045**	**0.746±0.0520**	**0.661±0.094**	**0.697±0.023**	**0.604±0.012**
PST-H(U)	0.635±0.026	0.644±0.050	0.635±0.054	0.645±0.061	**0.718±0.0647**	**0.644±0.044**	0.647±0.061	0.567±0.036
PPHTL	0.547±0.066	0.517±0.075	0.565±0.059	0.547±0.089	0.592±0.0806	0.615±0.071	0.558±0.065	0.524±0.027
PLR(target)	0.515±0.065	0.555±0.061	0.553±0.066	0.520±0.088	0.619±0.0701	0.563±0.026	0.558±0.060	0.517±0.053

Table 2: Testing AUC on all branches of RUIJIN data set. The best and comparable results according to pair-wise 95% significance test are high-lighted. Testing AUC of PLR on main center is 0.668±0.026.



(a) PST-F(U). (b) PST-F(W).

Figure 3: Testing AUC of low-levels models and different combining methods on NEWS20 ($\epsilon = 1.0$), where C-0 to C-4 are performance of low-level models.

name	importance	explaination
mchild	0.010	number of children
weight	0.012	birth weight
bone	0.013	bone mass measurement
eggw	0.005	frequency of having eggs
Glu120	0.055	glucose level 2 hours after meals
Glu0	0.060	glucose level immediately after meals
age	0.018	age
bmi	0.043	body mass index
HDL	0.045	high-density lipoprotein

Table 3: Some features in the RUIJIN data set, and importance is suggested by doctors. Top (resp. bottom) part: Features collected from the first (resp. second) investigation.

data set is collected by the Shanghai Ruijin Hospital during two investigations (in 2010 and 2013), conducted by the main hospital in Shanghai and 16 branches across China. The first investigation consists of questionnaires and laboratory tests collecting demographics, life-styles, disease information, and physical examination results. The second investigation includes diabetes diagnosis. Some collected features are shown in Table 3. Table 4 shows a total of 105,763 participants who appear in both two investigations. The smaller branches may not have sufficient labeled medical records for good prediction. Hence, it will be useful to borrow knowledge learned by the main hospital. However, users' privacy is a major concern, and patients' personal medical records in the main hospital should not be leaked to the branches.

In this section, we apply the method in Section 3.3 for diabetes prediction. Specifically, based on the patient data collected during the first investigation in 2010, we predict whether he/she will have diabetes diagnosed in 2013. The

main	#1	#2	#3	#4	#5	#6	#7	#8
12,702	4,334	4,739	6,121	2,327	5,619	6,360	4,966	5,793
	#9	#10	#11	#12	#13	#14	#15	#16
	6,215	3,659	5,579	2,316	4,285	6,017	6,482	4,493

Table 4: Number of samples collected from the main hospital and 16 branches in the RUIJIN data set.

main hospital serves as the source domain, and the branches are the target domains. We set $\epsilon_{\text{src}} = \epsilon_{\text{tgt}} = 1.0$. The following methods are also compared: (i) PLR(target), which directly uses PLR on the target data; (ii) PPHTL [Wang *et al.*, 2018]: a recently proposed privacy-preserving HTL method based on PLR; (iii) PST-F(U): There are 50 features, and they are randomly split into five groups, i.e., $K = 5$, and each group have equal weights; (iv) PST-F(W): Features are first sorted by importance, and then grouped as follows: The top 10 features are placed in the first group, the next 10 features go to the second group, and so on. q^k is set based on (6), with v_i being the importance values provided by the doctors. The other settings are the same as in Section 4.1.

Results are shown in Table 2. PPHTL may not have better performance than PLR(target), which is perhaps due to noise introduced in features. However, PST-F(U) improves over PPHTL by feature splitting, and consistently outperforms PLR(target). PST-F(W), which considers features importance, is the best.

5 Conclusion

In this paper, we propose a new privacy-preserving machine learning method, which improves privacy-preserving logistic regression by stacking. This can be done by either sample-based or feature-based partitioning of the data set. We provide theoretical justifications that the feature-based approach is better and requires a smaller sample complexity. Besides, when the importance of features is available, this can further boost the feature-based approach both in theory and practice. Effectiveness of the proposed method is verified on both standard benchmark data sets and a real-world cross-organizational diabetes prediction application.

Acknowledgments

We acknowledge the support of Hong Kong CERG-16209715. The first author also thanks Bo Han from Riken for helpful suggestions.

References

[Abadi *et al.*, 2016] M. Abadi, A. Chu, I. Goodfellow, B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *SIGSAC*, pages 308–318. ACM, 2016.

[Bassily *et al.*, 2014] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473. IEEE, 2014.

[Boyd *et al.*, 2011] S. Boyd, N. Parikh, and E. Chu. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[Breiman, 1996a] L. Breiman. Bagging predictors. *ML*, 24(2):123–140, 1996.

[Breiman, 1996b] L. Breiman. Stacked regressions. *ML*, 24(1):49–64, 1996.

[Chaudhuri *et al.*, 2011] K. Chaudhuri, C. Monteleoni, and A. Sarwate. Differentially private empirical risk minimization. *JMLR*, 12(Mar):1069–1109, 2011.

[Dwork and Roth, 2014] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Dwork *et al.*, 2006] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.

[Džeroski and Ženko, 2004] S. Džeroski and B. Ženko. Is combining classifiers with stacking better than selecting the best one? *ML*, 54(3):255–273, 2004.

[Emekçi *et al.*, 2007] F. Emekçi, O. Sahin, D. Agrawal, and A. El Abbadi. Privacy preserving decision tree learning over multiple parties. *TKDE*, 63(2):348–361, 2007.

[Fong and Weber-Jahnke, 2012] P. Fong and J. Weber-Jahnke. Privacy preserving decision tree learning using unrealized data sets. *TKDE*, 24(2):353–364, 2012.

[Friedman *et al.*, 2000] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[Friedman *et al.*, 2012] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer, 2012.

[Hamm *et al.*, 2016] J. Hamm, Y. Cao, and M. Belkin. Learning privately from multiparty data. In *ICML*, pages 555–563, 2016.

[Hanley and McNeil, 1983] J. Hanley and B. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

[Kasiviswanathan and Jin, 2016] P. Kasiviswanathan and H. Jin. Efficient private empirical risk minimization for high-dimensional learning. In *ICML*, pages 488–497, 2016.

[Kifer *et al.*, 2012] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. *JMLR*, 23, 2012.

[Kuzborskij and Orabona, 2013] I. Kuzborskij and F. Orabona. Stability and hypothesis transfer learning. In *ICML*, pages 942–950, 2013.

[Lang, 1995] Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*. Citeseer, 1995.

[LeCun *et al.*, 1998] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Ozay and Vural, 2012] M. Ozay and F. Vural. A new fuzzy stacked generalization technique and analysis of its performance. Technical report, arXiv:1204.0171, 2012.

[Pan and Yang, 2010] J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[Papernot *et al.*, 2017] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.

[Pathak *et al.*, 2010] M. Pathak, S. Rane, and B. Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *NIPS*, pages 1876–1884, 2010.

[Rajkumar and Agarwal, 2012] A. Rajkumar and S. Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *AISTAT*, pages 933–941, 2012.

[Shalev-Shwartz and Srebro, 2008] S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *ICML*, pages 928–935. ACM, 2008.

[Shokri and Shmatikov, 2015] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *SIGSAC*, pages 1310–1321, 2015.

[Smyth and Wolpert, 1999] P. Smyth and D. Wolpert. Linearly combining density estimators via stacking. *ML*, 36(1-2):59–83, 1999.

[Ting and Witten, 1999] K. Ting and I. Witten. Issues in stacked generalization. *JAIR*, 10:271–289, 1999.

[Wang *et al.*, 2018] Y. Wang, Q. Gu, and D. Brown. Differentially private hypothesis transfer learning. In *ECML*, 2018.

[Wolpert, 1992] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[Xie *et al.*, 2017] L. Xie, I. Baytas, K. Lin, and J. Zhou. Privacy-preserving distributed multi-task learning with asynchronous updates. In *SIGKDD*, pages 1195–1204, 2017.

[Zhou, 2012] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.