Pairwise Learning with Differential Privacy Guarantees

Mengdi Huai,[1]* Di Wang,[2] Chenglin Miao,[2] Jinhui Xu,[2] Aidong Zhang[1]
[1] Department of Computer Science, University of Virginia
[2]Department of Computer Science and Engineering, State University of New York at Buffalo
[1]{mh6ck, aidong}@virginia.edu, [2]{dwang45, cmiao, jinhui}@buffalo.edu

Abstract

Pairwise learning has received much attention recently as it is more capable of modeling the relative relationship between pairs of samples. Many machine learning tasks can be categorized as pairwise learning, such as AUC maximization and metric learning. Existing techniques for pairwise learning all fail to take into consideration a critical issue in their design, i.e., the protection of sensitive information in the training set. Models learned by such algorithms can implicitly memorize the details of sensitive information, which offers opportunity for malicious parties to infer it from the learned models. To address this challenging issue, in this paper, we propose several differentially private pairwise learning algorithms for both online and offline settings. Specifically, for the online setting, we first introduce a differentially private algorithm (called OnPairStrC) for strongly convex loss functions. Then, we extend this algorithm to general convex loss functions and give another differentially private algorithm (called OnPairC). For the offline setting, we also present two differentially private algorithms (called OffPairStrC and OffPairC) for strongly and general convex loss functions, respectively. These proposed algorithms can not only learn the model effectively from the data but also provide strong privacy protection guarantee for sensitive information in the training set. Extensive experiments on real-world datasets are conducted to evaluate the proposed algorithms and the experimental results support our theoretical analysis.

Introduction

As an important family of learning problems, *pairwise learning* has drawn much attention recently. Since pairwise learning involves a loss function depending on pairs of samples, it shows great advantage in modeling the relative relationship between pairs of samples over traditional pointwise learning (e.g., classification), in which the loss function only takes individual samples as the input. In practice, many learning tasks can be categorized as pairwise learning problmes. For instance, metric learning (Huai et al. 2019; Jin, Wang, and Zhou 2009; Huai et al. 2018a; Sun et al. 2012; Huai et al. 2018b; Suo et al. 2018) aims to learn a distance metric from a given collection of pair of similar/dissimilar samples that preserves the distance relation among the data, which can be formulated as a pairwise learning problem. Apart from metric learning, many other learning tasks, such as AUC maximization (Zhao et al. 2011; Natole, Ying, and Lyu 2018) and ranking (Tang and Wang 2018), can also be categorized as pairwise learning.

Existing pairwise learning algorithms can be roughly divided into two categories: *online* and *offline*. The online pairwise learning algorithms process the input data records in a sequential manner and iteratively update the model upon the arrival of each sample (Zhao et al. 2011; Kar et al. 2013). In contrast, the offline pairwise learning algorithms require the entire training dataset ready before the learning process starts and take it as whole to update the model (Cao, Guo, and Ying 2016; Jin, Wang, and Zhou 2009).

Although the importance of pairwise learning has been recognized in many real-world applications, existing pairwise learning algorithms fail to take into consideration an important issue in their designs, that is, the protection of sensitive information in the training set. The training datasets for pairwise learning are often collected from individual users and thus may contain private personal information. The models learned by such algorithms can implicitly memorize some details of the sensitive information, which undesirably offers opportunity for malicious parties to compromise the users' privacy. Taking the patient similarity learning task as example, a hospital may want to train a universal patient similarity learning model from patients (crossing many hospitals) so as to obtain a better understanding of the diseases and diagnoses. Due to trust to the hospital, patients may be willing to provide necessary information for such a learning process. However, without a proper mechanism, the patients' privacy may be breached when the trained model by the hospital is provided to other parties (such as medical research institutes or drug makers). This is because these parties can infer patients' private information using various attack techniques, such as model inversion attack (Fredrikson, Jha, and Ristenpart) and membership attack (Shokri et al. 2017). Thus, without a convincing privacy-preserving mechanism, the patients may not be willing to participate in such a learning task. Hence, a big challenge facing pairwise learning is how to learn a model privately such that sensitive information cannot be inferred from the learned model.

To the best of our knowledge, no existing work has addressed the above challenge. This motivates us to design,

[*] The first two authors contributed equally to this work.

in this paper, privacy-preserving pairwise learning methods which can not only keep the sensitive information private but also guarantee good generalization performance. Among existing privacy-preserving strategies, differential privacy (DP) (Dwork et al. 2006), as a rigorous notion for data privacy, can provide very rigid privacy and utility guarantee. Although various DP methods exist for (online) pointwise learning, such as objective perturbation or DP-SGD (Chaudhuri and Monteleoni 2009; Bassily, Smith, and Thakurta 2014; Jain, Kothari, and Thakurta 2012; Wang, Chen, and Xu 2019), they cannot be applied to pairwise learning algorithms directly. This is mainly because the training sample pairs in pairwise learning algorithms are not i.i.d. and the loss function depends on more than one data records. In the light of the above challenges, in this paper, we propose efficient differentially private algorithms for the aforementioned two types of pairwise learning problems. The contributions of this paper can be summarized as follows:

- Firstly, we consider the pairwise learning problem in the online setting, and propose an (ϵ, δ)-DP algorithm called <u>on</u>line <u>pair</u>wise private GIGA-<u>Str</u>ongly <u>c</u>onvex method (**OnPairStrC**). This algorithm achieves a regret upper bound of $\tilde{O}(\frac{\sqrt{d}\sqrt{n}}{\epsilon})$ when the losses are strongly convex, where d is the feature dimension and n is the data size. We then extend this algorithm to general convex losses by proposing an algorithm called <u>on</u>line <u>pair</u>wise private GIGA-<u>c</u>onvex method (**OnPairC**), which has a regret upper bound of $\tilde{O}(\frac{\sqrt{d}n^{\frac{3}{4}}}{\epsilon})$.
- Secondly, we study the pairwise learning problem in the offline setting. We show that it is possible to achieve generalization errors of $\tilde{O}(\frac{\sqrt{d}}{\sqrt{n}\epsilon})$ and $\tilde{O}(\frac{\sqrt{d}}{\sqrt[4]{n}\epsilon})$ for strongly and general convex loss functions respectively by adopting the results in the online settings. We then improve these bounds by proposing an <u>off</u>line <u>pair</u>wise private GIGA-<u>Str</u>ongly <u>c</u>onvex algorithm (**OffPairStrC**) and an <u>off</u>line <u>pair</u>wise private GIGA-<u>c</u>onvex algorithm (**OffPairC**) for the two types of loss functions. Particularly, in the case of general convex loss functions, our improved algorithm can achieve a generalization error of $\tilde{O}(\frac{\sqrt{d}}{\sqrt{n}\epsilon})$.

Related Work

As mentioned earlier, there is no existing result on pairwise learning under the differential privacy model. Thus, we only compare ours with those private pointwise learning methods. There is a long list of papers on differentially private pointwise learning in the last decade which attack the problem from different perspectives. For DP pointwise learning with convex loss functions, there are a lot of works on it, such as (Chaudhuri and Monteleoni 2009; Chaudhuri, Monteleoni, and Sarwate ; Bassily, Smith, and Thakurta 2014; Wang, Ye, and Xu 2017; Wang, Chen, and Xu 2019). However, all of the above results focus only on pointwise loss functions and cannot be extended to pairwise loss functions.

Differentially private pointwise learning in the online setting has also been studied previously (Jain, Kothari, and Thakurta 2012; Thakurta and Smith 2013). The works that are most related to ours are probably (Jain, Kothari, and Thakurta 2012) and (Thakurta and Smith 2013), where the authors gave a general framework for online convex optimization under differential privacy. However, there are some significant differences from ours. Firstly, (Jain, Kothari, and Thakurta 2012) and (Thakurta and Smith 2013) consider only pointwise loss functions while we study pairwise loss functions. Thus, their methods cannot be directly extended to pairwise loss functions, making them incomparable with ours; secondly, due to the differences in the structure of two problems and the definitions of the regret, the analyses of the upper bounds and the DP guarantees are quite different (see Remark 1 for more details).

Preliminaries

We say that two datasets D, D' are neighbors if they differ by only one entry, which is denoted as $D \sim D'$.

Definition 1 (Differential Privacy (Dwork et al. 2006))**.** *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private (DP) if for all neighboring datasets D, D' and for all events S in the output space of $\mathcal{A}$, the following holds*

$$Pr(\mathcal{A}(D) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(D') \in S) + \delta.$$

When $\delta = 0$, $\mathcal{A}$ is ϵ-differentially private.

In this paper we focus on (ϵ, δ)-DP and use Gaussian mechanism (Dwork et al. 2006) to guarantee (ϵ, δ)-DP.

Definition 2 (Gaussian Mechanism)**.** *Given any function $q : \mathcal{X}^n \to \mathbb{R}^d$, the Gaussian mechanism is defined as $\mathcal{M}_G(D, q, \epsilon) = q(D) + Y$, where Y is drawn from Gaussian Distribution $\mathcal{N}(0, \sigma^2 I_d)$ with $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}\Delta_2(q)}{\epsilon}$. Here $\Delta_2(q)$ is the ℓ_2-sensitivity of the function q, i.e., $\Delta_2(q) = \sup_{D \sim D'} ||q(D) - q(D')||_2$. Gaussian mechanism preserves (ϵ, δ)-differential privacy.*

Additionally, we also use zero Concentrated Differential Privacy (zCDP) (Bun and Steinke 2016) and its composition property to guarantee (ϵ, δ)-DP. Compared to directly using the composition property of DP, it has many advantages (see (Lee and Kifer 2018; Wang and Xu 2019) for more details).

Definition 3. *A randomized mechanism $\mathcal{A}$ is ρ-zCDP if, for all neighboring dataset D, D' and all $\alpha \in (1, \infty)$,*

$$D_{\alpha}(\mathcal{A}(D)||\mathcal{A}(D')) \leq \rho\alpha,$$

where $D_{\alpha}(\cdot||\cdot)$ is the α-Rényi Divergence [1].

Lemma 1 ((Bun and Steinke 2016))**.** *Suppose that two mechanisms satisfy ρ_1-zCDP and ρ_2-zCDP, respectively. Then, their composition is $(\rho_1 + \rho_2)$-zCDP.*

Lemma 2 ((Bun and Steinke 2016))**.** *For a Gaussian mechanism $q(D) + Y$ with $Y \sim \mathcal{N}(0, \sigma^2 I_d)$, it satisfies $(\frac{\Delta_2^2(q)}{2\sigma^2})$-zCDP.*

Lemma 3 ((Bun and Steinke 2016))**.** *If a mechanism is ρ-zCDP, then it is $(\rho + 2\sqrt{\rho \log \frac{1}{\delta}}, \delta)$-DP for any $\delta > 0$.*

[1] For two distributions P and Q on Ω and $\alpha \in (1, \infty)$, the α-Rényi Divergence between P, Q is defined as $D_{\alpha}(P||Q) = \frac{1}{\alpha-1} \log \int_{\Omega} P(x)^{\alpha} Q(x)^{1-\alpha} dx$.

Private Pairwise Learning

Different from the pointwise loss function $\ell : \mathcal{C} \times \mathcal{D} \mapsto \mathbb{R}$, a pairwise loss function is a function on pairs of data records, *i.e.,* $\ell : \mathcal{C} \times \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}$, where $\mathcal{D}$ is the data universe. Given a dataset $D = \{z_1, z_2, \cdots, z_n\} \subseteq \mathcal{D}^n$ and a loss function $\ell(\cdot; \cdot, \cdot)$, its empirical risk can be defined as:

$$L(w; D) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \ell(w; z_i, z_j). \tag{1}$$

When the inputs are drawn i.i.d from an unknown underlying distribution $\mathcal{P}$ on $\mathcal{D}$, the population risk is

$$L_{\mathcal{P}}(w) = \mathbb{E}_{z_i, z_j \sim \mathcal{P}}[\ell(w; z_i, z_j)]. \tag{2}$$

Similar to the definition of private pointwise learning (Bassily, Smith, and Thakurta 2014), we can define private pairwise learning as follows.

Definition 4 (Private pairwise learning). *Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a convex, closed and bounded constraint set, $\mathcal{D}$ be a data universe, and $\ell : \mathcal{C} \times \mathcal{D} \times \mathcal{D} \mapsto \mathbb{R}$ be a pairwise loss function. Also, let $D = \{z_1 = (x_1, y_1), z_2 = (x_2, y_2), \cdots, z_n = (x_n, y_n)\} \subseteq \mathcal{D}^n$ be a dataset with data records $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$ and labels (responses) $\{y_i\}_{i=1}^n \subset [-1, 1]^n$. Private pairwise learning is to find a private estimator $w_{priv} \in \mathcal{C}$ so that the algorithm is (ϵ, δ) or ϵ differential privacy and the error is minimized, where the error for an estimator w can be measured by either the optimality gap $Err_D(w) = L(w; D) - \min_{w \in \mathcal{C}} L(w; D)$ or the generalization error $Err_{\mathcal{P}}(w) = L_{\mathcal{P}}(w) - \min_{w \in \mathcal{C}} L_{\mathcal{P}}(w)$.*

In this paper, we will focus on a special class of pairwise loss functions [2] which contains the loss functions of metric learning, AUC maximization and bipartite ranking.

Assumption 1. *For the loss function, we assume that it has the form of $\ell(w; z, z') = \phi(Y(y, y')h(w; x, x'))$, and ℓ is a G-Lipschitz and L-smooth convex function over w, where $Y(y, y') = y - y'$ or $Y(y, y') = yy'$. In the experimental part, we will let ϕ be the logistic function, i.e., $\phi(x) = \log(1 + e^{-x})$.*

Example 1: Metric Learning (Cao, Guo, and Ying 2016) The goal here is to learn a Mahalanobios metric $M_W^2(x, x') = (x - x')^T W(w - x')$ using loss function $\ell(W; z, z') = \phi(yy'(1 - M_W^2(x, x'))$, where $y, y' \in \{-1, +1\}$. The constraint set $\mathcal{C}$ is $\mathcal{C} = \{W : W \in \mathbb{S}^d, \|W\|_F \leq 1\}$.

Example 2: AUC Maximization (Zhao et al. 2011), Bipartite Ranking (Clémençon et al. 2008) The goal here is to maximize the area under the ROC curve for a linear classification problem with the constraint of $\|w\|_2 \leq 1$. Here $h(w; x, x') = w^T(x - x')$ and $\ell(w; z, z') = \phi((y - y')h(w; x, x'))$, where $y, y' \in \{-1, +1\}$.

Online Private Pairwise Learning

Here we follow online pairwise learning in (Kar et al. 2013). An online learning algorithm $\mathcal{A}$ is given sequential access to a stream of elements $z_1, z_2, z_3, \cdots, z_n$. At each time step $t = 2, 3, \cdots, n$, the algorithm selects a parameter $w_{t-1} \in \mathcal{C}$ upon which the data record z_t is revealed, and the algorithm incurs the following penalty

$$\hat{L}_t(w_{t-1}, D_t) = \frac{1}{t-1} \sum_{i=1}^{t-1} \ell(w_{t-1}; z_t, z_i), \tag{3}$$

where $D_t = \{z_1, \cdots, z_t\}$. Thus, the online algorithm $\mathcal{A}$ maps a data sequence $\{z_1, \cdots, z_n\}$ to a sequence of parameters $\{w_1, \cdots, w_{n-1}\}$. In the non-private case, the goal is to select $\{w_1, \cdots, w_{n-1}\}$ so as to minimize the **regret**, *i.e.,*

$$\mathcal{R}_{\mathcal{A}}(n, D) = \sum_{t=2}^{n} \hat{L}_t(w_{t-1}, D_t) - \min_{w \in \mathcal{C}} \sum_{t=2}^{n} \hat{L}_t(w, D_t). \tag{4}$$

Moreover, if all data are chosen i.i.d from the distribution $\mathcal{P}$, we also want to minimize the **generalized regret**, *i.e.,*

$$\mathcal{R}_{\mathcal{P},\mathcal{A}}(n) = \sum_{t=2}^{n} L_{\mathcal{P}}(w_{t-1}) - (n-1) \min_{w \in \mathcal{C}} L_{\mathcal{P}}(w). \tag{5}$$

If ℓ is convex, then from (5) we have parameter $\bar{w} = \frac{w_1 + \cdots + w_{n-1}}{n-1}$ satisfies the following generalization error:

$$L_{\mathcal{P}}(\bar{w}) - \min_{w \in \mathcal{C}} L_{\mathcal{P}}(w) \leq \frac{\mathcal{R}_{\mathcal{P},\mathcal{A}}(n)}{n-1}. \tag{6}$$

However, under the differential privacy model, we need to guarantee that the output sequence $\{w_1, \cdots, w_{n-1}\}$ is DP. Thus, private pairwise learning in the online setting can be defined as follows:

Definition 5 (Online private pairwise learning). *Let $Z = \{z_1, z_2, \cdots, z_n\}$ be any sequence of data records in the data universe $\mathcal{D}$. Let the sequence of outputs by algorithm $\mathcal{A}$ be $\mathcal{A}(Z) = \{w_1, \cdots, w_{n-1}\}$. Then, $\mathcal{A}$ is (ϵ, δ)-differentially private if given any other data sequence Z' which differs in at most one entry with Z, for all events S, we have $Pr[\mathcal{A}(Z) \in S] \leq e^\epsilon Pr[\mathcal{A}(Z') \in S] + \delta$. The goal of online private pairwise learning is to select private outputs $\{w_1, \cdots, w_{n-1}\}$ that minimizes the (generalized) regret.*

From above discussions on (5) and (6), we know that if the generalized regret is low, the algorithm will have a good performance on generalization theoretically. From this view, the online setting is more general. Thus, in the paper, we will first consider the online private pairwise learning and provide regrets for both strongly and general convex losses. After that, we will study the problem in the offline setting.

Online Private Pairwise Learning

We first consider the case that the loss function is strongly convex. After that, we will use the regularization perturbation strategy in (Thakurta and Smith 2013) to extend the resulting algorithm to general convex loss functions.

Our algorithm is inspired by the stability of Generalized Infinitesimal Gradient Ascent (GIGA) (Zinkevich 2003; Jain, Kothari, and Thakurta 2012), which is a well-known online convex algorithm (see Remark 1 for discussions on

[2] We note that all the (ϵ, δ)-DP algorithms in this paper can be extended to general pairwise loss functions, although the upper bounds of the generalization errors may differ.

Algorithm 1 Online Pairwise Private GIGA-Strongly Convex (OnPairStrC)

```
1:  **Input:** Privacy parameters $\epsilon$ and $\delta$, sequence of data record $\{z_1, z_2, \cdots, z_n\}$, constrained convex set $\mathcal{C} \subset \mathbb{R}^d$, and pairwise loss function $\ell(\cdot;\cdot,\cdot)$.
2:  **Parameters:** $\ell$ is $G$-Lipschitz, $L$-smooth and $\alpha$-strongly convex over $w$. Step time $T_1 = \max\{\lceil \frac{16L^2}{\alpha^2} \rceil, 7\}$.
3:  Compute $\rho$ which satisfies $\rho + 2\sqrt{\rho \log(\frac{1}{\delta})} = \epsilon$.
4:  **for** $t = 1, \cdots, T_1$ **do**
5:      Receive the data record $z_t$ (incurs penalty $\hat{L}_t(w_{t-1}, D_t)$ when $t \geq 2$).
6:      Randomly choose a parameter $w_t \in \mathcal{C}$.
7:  **end for**
8:  **for** $t = T_1 + 1, \cdots, n$ **do**
9:      Receive the data record $z_t$ (incurs penalty $\hat{L}_t(w_{t-1}, D_t)$).
10:     Set step size $\eta_t = \frac{t-1}{t-2}\frac{2}{\alpha t}$
11:     $w_t = \Pi_{\mathcal{C}}[w_{t-1} - \eta_t \nabla \hat{L}_t(w_{t-1}, D_t)]$, where $\Pi_{\mathcal{C}}$ is the projection onto the set $\mathcal{C}$.
12:     Set $\sigma_t^2 = \frac{32G^2(n-T_1)}{\alpha^2 t^2 \rho}$. Let $\tilde{w}_t = w_t + n_t$, where $n_t \sim \mathcal{N}(0, \sigma_t^2 I_d)$.
13:     Output $w_t = \arg\min_{w \in \mathcal{C}} \|w - \tilde{w}_t\|_2^2$.
14: **end for**
```

the difference of our algorithm with the previous ones). The main steps are given in Algorithm 1.

We call the above algorithm excluding the portion of random perturbation (*i.e.,* steps 12 and 13) **Pairwise GIGA**. The following lemma gives an upper bound on the ℓ_2-norm sensitivity of the output in the t-th iteration of Pairwise GIGA, which is to ensure (ϵ, δ)-DP of Algorithm 1.

Lemma 4. *Let $\mathcal{A}_t(D_t)$ denote the output of Pairwise GIGA in the t-th iteration. Then, under the assumption of Algorithm 1, for any $t \geq 1$ and $D_t \sim D'_t$,*

$$\|\mathcal{A}_t(D_t) - \mathcal{A}_t(D'_t)\|_2 \leq \frac{8G}{\alpha t}.$$

In Algorithm 1, steps 4 to 7 seem weird due to the random sampling of w_{t-1}. However, as we see from the proof of Lemma 4, this condition is necessary for the stability analysis. Moreover, the similar steps have also been adopted in some algorithms on DP online learning, such as (Thakurta and Smith 2013; Jain, Kothari, and Thakurta 2012).

Theorem 1. *Under Assumption 1 and the assumption that the loss function ℓ is α-strongly convex, for any $0 < \epsilon, \delta \leq 1$, Algorithm 1 is (ϵ, δ)-differentially private.*

Note that to guarantee DP, we first transfer (ϵ, δ)-DP to ρ-zCDP by Lemma 3, and then use composition theorem to make Algorithm 1 be ρ-zCDP (*i.e.,* we make each iteration $T_1 + 1 \leq t \leq n$ be $\frac{\rho}{n-T_1}$-zCDP). It is easy to see that in this case the variance of the noise satisfies $\sigma_t^2 = \frac{32G^2(n-T_1)}{\alpha^2 t^2 (\sqrt{\log(1/\delta)+\epsilon} - \sqrt{\log(1/\delta)})^2}$. When $\frac{\epsilon}{\log(1/\delta)} \ll 1$ (this case will always holds since in practice we select $\epsilon = 0.1 \sim 5$ and $\delta = \frac{1}{n}$), by Taylor expansion of $\sqrt{1+x}$, we have $(\sqrt{\log(1/\delta)+\epsilon} - \sqrt{\log(1/\delta)})^2 \simeq \frac{\epsilon^2}{4\log(1/\delta)}$. Thus in total, we have $\sigma_t^2 \simeq \frac{128G^2(n-T_1)\log(1/\delta)}{\alpha^2 t^2 \epsilon^2}$.

The following theorem shows an upper bound on the regret of Algorithm 1, which can be transformed to generalized error (we will show it in the following section).

Theorem 2. *Under the assumptions in Theorem 1 and the additional condition of $\frac{\epsilon}{\log\frac{1}{\delta}} \ll 1$, Algorithm 1 has the following upper bound on the regret of its outputs*

$$\mathcal{R}_{\mathcal{A}}(n, D) \leq O(\frac{G^2\sqrt{d}\log^{1.5}\frac{n}{\zeta}\sqrt{n}\sqrt{\log\frac{1}{\delta}}}{\alpha\epsilon} + \frac{GL^2}{\alpha^2}\|\mathcal{C}\|_2 + \frac{G^2\log n}{\alpha}) \quad (7)$$

with probability at least $1 - \zeta$. where $\|\mathcal{C}\|_2 = \max_{w,w' \in \mathcal{C}} \|w - w'\|_2$ is the diameter of the set $\mathcal{C}$[3]*.*

Remark 1. *We note that (Jain, Kothari, and Thakurta 2012) used the differentially private version of GIGA and IGD (Kulis and Bartlett) in their DP pointwise learning. But their Private GIGA or IGD algorithm is quite different from our method of OnPairStrC (Algorithm 1). Firstly, (Jain, Kothari, and Thakurta 2012) needs to assume that each loss function $\hat{L}_t$ is independent (see the proofs of Lemma 4 and Lemma 5 in (Jain, Kothari, and Thakurta 2012)), which means that it is only applicable to pointwise loss functions. However, in our problem, the penalty function (3) depends on previous data records, which means that it is much more complicated than the case in (Jain, Kothari, and Thakurta 2012). Thus, we need a much finer and more different analysis on the stability of Pairwise GIGA. Also, the parameters of the step size η_t and time step T_1 are quite different from those in (Jain, Kothari, and Thakurta 2012) (see the supplementary material for details). Additionally, in order to show the power of our method, we also consider the case with additional finite buffer constraint, which has not been studied in (Jain, Kothari, and Thakurta 2012) (see the supplementary material for details). Thus, our method is more general.*

Secondly, the upper bound (7) on the regret of our algorithm is less than that in (Jain, Kothari, and Thakurta 2012) with a factor of $\log\frac{n}{\delta}$. This is due to the fact that we use the composition property of zCDP instead of advanced composition theorem of DP (Dwork, Rothblum, and Vadhan 2010).

*Thirdly, since the definition of regret in our paper is different from that in pointwise learning (Jain, Kothari, and Thakurta 2012), the same upper bound (*i.e., *$\tilde{O}(\frac{\sqrt{dn}}{\epsilon})$) on the regret for strongly convex losses are actually incomparable.*

We now use the perturbation strategy in (Thakurta and Smith 2013) to obtain result for general convex losses.

Theorem 3. *Let ℓ be a pairwise loss function satisfying Assumption 1. Then, for any $0 < \epsilon, \delta \leq 1$, Algorithm 2 is (ϵ, δ)-DP. Moreover, if $\frac{\epsilon}{\log\frac{1}{\delta}} \ll 1$ and take $\alpha = O(\frac{1}{\sqrt[4]{n}})$, then with*

[3] If $\mathcal{C} = \mathbb{R}^d$, then we can take $\mathcal{C} = \{w : \|w\|_2 \leq \|w^*\|_2\}$.

Algorithm 2 Online Pairwise Private GIGA-Convex (OnPairC)

```
Input: Privacy parameters $\epsilon$ and $\delta$, sequence of data record $\{z_1, z_2, \cdots, z_n\}$, constrained convex set $\mathcal{C}$, pairwise loss $\ell(\cdot;\cdot,\cdot)$, and a parameter $\alpha$ to be defined later.
Parameters: $\ell$ is $G$-Lipschitz, $L$-smooth and convex over $w$.
Randomly select a point $w_0 \in \mathcal{C}$. Let $\tilde{\ell}(w;z,z') = \ell(w;z,z') + \frac{\alpha}{2}\|w - w_0\|_2^2$.
Run Algorithm 1 with loss $\tilde{\ell}$, which is $\tilde{G} = G + \alpha\|\mathcal{C}\|_2$-Lipschitz, $\tilde{L} = L + \alpha$-smooth and $\alpha$-strongly convex.
```

probability at least $1 - \zeta$, the following upper bound on regret for the outputs holds:

$$\mathcal{R}_{\mathcal{A}}(n, D) \le O(\frac{L^2G^2\|\mathcal{C}\|_2^2\sqrt{d}\log^{1.5}\frac{n}{\zeta}n^{\frac{3}{4}}\sqrt{\log\frac{1}{\delta}}}{\epsilon}). \quad (8)$$

Comparing (8) with (7), we can see that for strongly convex pairwise loss functions, the average regret, *i.e.,* $\frac{\mathcal{R}_{\mathcal{A}}(n)}{n-1}$, is upper bounded by $\tilde{O}(\frac{\sqrt{d}}{\sqrt{n}\epsilon})$, while for general convex ones, it is $\tilde{O}(\frac{\sqrt{d}}{\sqrt[4]{n}\epsilon})$. This is the same as in the case of pointwise loss functions (Thakurta and Smith 2013).

Offline Private Pairwise Learning

Generalization Error Induced by Generalized Regret

We first observe that Algorithm 1 and 2 preserve (ϵ, δ)-DP in the offline settings. Also, as discussed in (5) and (6), if we get the generalized regret for the output $\{w_1, w_2, \cdots, w_{n-1}\}$, we can easily obtain a generalization error by (6). By a theorem in (Kar et al. 2013), we can have the following generalization bounds for $\bar{w} = \frac{w_1+\cdots+w_{n-1}}{n-1}$ of Algorithm 1 and 2. Before this, we first let the Rademacher averages of the pairwise loss functions class $\ell \circ \mathcal{C} := \{(z, z') \mapsto \ell(w; z, z'), w \in \mathcal{C}\}$ be denoted by the following (Kar et al. 2013):

$$\mathcal{R}_n(\ell \circ \mathcal{C}) = \mathbb{E}[\sup_{w\in\mathcal{C}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \ell(w; z, z_i)], \quad (9)$$

where $\{\epsilon_i\}_{i=1}^n$ are the Rademacher variables, and the expectation is over $\{\epsilon_i\}_{i=1}^n, z, \{z_i\}_{i=1}^n$.

Theorem 4. *Under Assumption 1, the parameter $\bar{w} = \frac{w_1+\cdots+w_{n-1}}{n-1}$ satisfies the following generalization error for loss function ℓ with probability at least $1 - 2\zeta$ if $\frac{\epsilon}{\log\frac{1}{\delta}} \ll 1$, where $w_1, w_2, \cdots, w_{n-1}$ are the outputs of Algorithm 2 (Algorithm 1 for strongly convex loss functions),*

$$Err_{\mathcal{P}}(\bar{w}) \le O\big(\frac{\sum_{t=2}^n \mathcal{R}_{t-1}(\ell \circ \mathcal{C})}{n-1} + \frac{L^2G^2\|\mathcal{C}\|_2^2\sqrt{d}\log^{1.5}\frac{n}{\zeta}\sqrt{\log\frac{1}{\delta}}}{\epsilon\sqrt[4]{n}}\big). \quad (10)$$

Moreover, if the loss is α-strongly convex, then we have:

$$Err_{\mathcal{P}}(\bar{w}) \le O\big(\frac{1}{n-1}\sum_{t=2}^{n}\mathcal{R}_{t-1}(\ell \circ \mathcal{C}) + \frac{G^2L^2\|\mathcal{C}\|_2\sqrt{d}\log^{1.5}\frac{n}{\zeta}\sqrt{\log\frac{1}{\delta}}}{\alpha^2\epsilon\sqrt{n}}\big). \quad (11)$$

Remark 2. *Note that there are many problems whose Rademacher average is $\mathcal{R}_n(\ell \circ \mathcal{C}) = O(\frac{\sqrt{d}}{\sqrt{n}})$, e.g. Example 1 and 2 (Kar et al. 2013). Thus for Example 1, the generalization error is $\tilde{O}(\frac{d}{\epsilon\sqrt[4]{n}})$ for logistic loss while it is $\tilde{O}(\frac{d}{\epsilon\sqrt{n}})$ if adding an additional Frobenious regularization to the losses. Similar result holds for Example 2, where the generalization error is $\tilde{O}(\frac{\sqrt{d}}{\epsilon\sqrt[4]{n}})$ while it is $\tilde{O}(\frac{\sqrt{d}}{\epsilon\sqrt{n}})$ in the case of with additional ℓ_2-norm regularization.*

Improved Upper Bounds by Offline Differentially Private Algorithms

Inspired by the sensitivity of Pairwise GIGA in Lemma 4 and Theorem 4, we propose an offline DP algorithm which has better upper bounds compared to (10) and (11). The basic idea is to use output perturbation. Specifically, we first run Pairwise GIGA in the offline settings and then add some Gaussian noises to $\tilde{w} = \frac{w_1+\cdots+w_n}{n}$ to keep the algorithm (ϵ, δ)-DP, since the sensitivity of $\tilde{w}$ is based on each w_i, which can be obtained by Lemma 4. For general convex loss functions, we can still use the perturbation idea, which is the same as in Algorithm 2. See Algorithm 3 and 4 for details.

The reason that we can improve the generalization error is due to the following fact. From Algorithms 1 and 2, we can see that the output sequences $\{w_1, w_2, \cdots, w_{n-1}\}$ satisfy the conditions of (ϵ, δ)-DP in each iteration. However, in the offline setting, we only need to ensure that the final output is DP. Thus, instead of adding noise in each iteration, we can add noises only once to the final output, meaning that we can add a smaller scale of noises compared to the online ones.

Theorem 5. *For any $0 < \epsilon, \delta \le 1$, Algorithm 3 is (ϵ, δ)-DP for any α-strongly convex loss functions satisfying Assumption 1. Moreover, if $\frac{\epsilon}{\log\frac{1}{\delta}} \ll 1$, then with probability at least $1 - 2\zeta$, the output $\hat{w}$ satisfies:*

$$Err_{\mathcal{P}}(\hat{w}) \le O(\frac{\sqrt{d}G^2\|\mathcal{C}\|_2\log\frac{n}{\zeta}\sqrt{\log\frac{1}{\delta}}}{\alpha n\epsilon} + \frac{1}{n}\sum_{t=1}^{n}\mathcal{R}_t(\ell \circ \mathcal{C})). \quad (12)$$

Algorithm 4 is (ϵ, δ)-DP for any convex loss functions satisfying Assumption 1 if $\alpha = O(\frac{1}{\sqrt{n}})$. Moreover, if $\frac{\epsilon}{\log\frac{1}{\delta}} \ll 1$, then with probability at least $1 - 2\zeta$, the output $\hat{w}$ satisfies:

$$Err_{\mathcal{P}}(\hat{w}) \le O\big(\frac{\sqrt{d}G^2\|\mathcal{C}\|_2^2\log\frac{n}{\zeta}\sqrt{\log\frac{1}{\delta}}\log n}{\sqrt{n}\epsilon} + \frac{1}{n}\sum_{t=1}^{n}\mathcal{R}_t(\ell \circ \mathcal{C})\big). \quad (13)$$

From Theorem 5, we can see that for strongly and general convex loss functions, the bounds in (13) and (12) are respectively lower than those in (10) and (11). Specifically, for general convex loss functions, we can improve the upper bound from $\tilde{O}(\frac{\sqrt{d}}{\epsilon \sqrt[4]{n}})$ to $\tilde{O}(\frac{\sqrt{d}}{\epsilon\sqrt{n}})$ if $\mathcal{R}_n(\ell \circ \mathcal{C}) = O(\frac{\sqrt{d}}{\sqrt{n}})$.

Algorithm 3 Offline Pairwise Private GIGA-Strongly Convex (OffPairStrC)

```
**Input:** Privacy parameters $\epsilon$ and $\delta$, sequence of data $\{z_1, z_2, \cdots, z_n\}$, constrained convex set $\mathcal{C}$, pairwise loss $\ell(\cdot;\cdot,\cdot)$, and step number $T_1 = \max\{\lceil \frac{16L^2}{\alpha^2} \rceil, 7\}$.
**Parameters:** $\ell$ is $G$-Lipschitz, $L$-smooth and $\alpha$-strongly convex over $w$.
Randomly sample $w_1, \cdots, w_{T_1} \in \mathcal{C}$.
**for** $t = T_1 + 1, \cdots, n$ **do**
    Set step size $\eta_t = \frac{t-1}{t-2}\frac{2}{\alpha t}$.
    Update
        $$w_t = \arg\min_{w \in \mathcal{C}} \|w - (w_{t-1} - \eta_t \nabla \hat{L}_t(w_{t-1}, D_t))\|_2^2.$$
**end for**
Let $\tilde{w} = \frac{w_1 + \cdots + w_n}{n}$.
Denote $\bar{w} = \tilde{w} + \sigma$, where $\sigma \sim \mathcal{N}(0, \frac{128G^2 \log^2 n \log(1.25/\delta)}{\alpha^2 n^2 \epsilon^2} I_d)$.
Return $\hat{w} = \arg\min_{w \in \mathcal{C}} \|w - \bar{w}\|_2^2$.
```

Algorithm 4 Pairwise Private GIGA-Convex (OffPairC)

```
**Input:** Privacy parameters $\epsilon$ and $\delta$, sequence of data $\{z_1, \cdots, z_n\}$, constrained convex set $\mathcal{C}$, pairwise loss function $\ell(\cdot;\cdot,\cdot)$, and a parameter $\alpha$ to be defined later.
**Parameters:** $\ell$ is $G$-Lipschitz, $L$-smooth and convex over $w$.
Let $\tilde{\ell}(w; z, z') = \ell(w; z, z') + \frac{\alpha}{2}\|w - w_0\|_2^2$, $w_0$ is any point in $\mathcal{C}$.
Run Algorithm 3 with loss $\tilde{\ell}$, which is $\tilde{G} = G + \alpha\|\mathcal{C}\|_2$-Lipschitz, $\tilde{L} = L + \alpha$-smooth and $\alpha$-strongly convex.
```

Experiments

In this section, we empirically evaluate the performance of the proposed differentially private algorithms on real-world datasets. we take two popular pairwise learning tasks, i.e., AUC maximization and metric learning, as examples. All of the experiments in this paper are conducted over 20 runs of different random permutations for each adopted dataset, and we report the averaged results.

Experimental Setup

Datasets. We use four real-world datasets that are widely adopted in pairwise learning tasks. These datasets are the **Diabetes** dataset, the **Diabetic Retinopathy** dataset, the **Hepatitis** dataset and the **Cancer** dataset (Dua and Graff 2017).
Performance measures. To evaluate the performance of the proposed algorithms, we use the following measures:

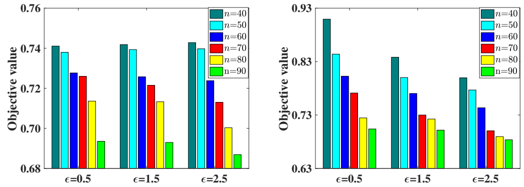
1. *AUC*: For AUC maximization task, we report the AUC measurement (Zhao et al. 2011) for each of the proposed algorithms over every adopted dataset. A larger AUC value means that the corresponding AUC maximization algorithm can generate more accurate results.
2. *Classification Accuracy*: For metric learning task, we calculate the classification accuracy that is defined as the percentage of the correctly classified samples in the test set. The less the classification accuracy, the worse the performance of the proposed algorithm. In this paper, the KNN classifier is adopted to assign labels to the test samples. For the KNN classifier, we set K to be 3.
3. *Objective function value*: For both metric learning task and AUC maximization task, we also report the objective function value of the proposed differentially private algorithms. A smaller objective function value means that the original pairwise learning model is less perturbed.

Baselines. Since there is no existing work that addresses the privacy issue in pairwise learning, in experiments, we take the original pairwise learning algorithms that do not take any actions to protect the private information as the baselines. We denote the baseline methods as **NonPrivate**, which is the GIGA for pairwise loss functions (Kar et al. 2013).

Experiments for AUC maximization

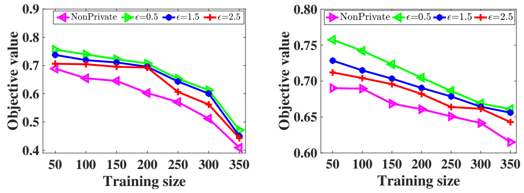
We first evaluate the performance of the proposed differentially private pairwise learning algorithms (i.e., OnPairStrC, OnPairC, OffPairStrC and OffPairC) for AUC maximization task (see Example 2 for the problem formulation). We add additional ℓ_2 regularization $\frac{\lambda}{2}\|w\|_2^2$ with $\lambda = 10^{-3}$ to loss function for the strongly convex case.

We study the effect of the training size n and the privacy parameter ϵ on the performance of the proposed OnPairStrC, OnPairC, OffPairStrC and OffPairC algorithms. Here we fix $\delta = \frac{1}{n}$ and consider three cases where the value of parameter ϵ is set to be 0.5, 1.5 and 2.5, respectively. For OnPairStrC and OffPairStrC, we vary the training size from 40 to 90 and conduct the experiment on the Hepatitis and Cancer datasets. For OnPairC and OffPairC, the experiment is conducted on the Diabetes and Diabetic Retinopathy datasets and we vary the training size from 50 to 350. In Figure 1 and Figure 2, we respectively report the objective values of OnPairStrC and OnPairC. The experimental results show that the larger the value of the training size n, the smaller the objective value. Additionally, when n is fixed, the smaller the value of ϵ, the larger the objective value is. The performance of the proposed algorithms are comparable with that of the baseline, which can be observed from Figure 2. The results for OffPairStrC and OffPairC are shown in Figure 3 and Figure 4, respectively. Figure 3 shows the objective value of OffPairStrC when the training size varies and Figure 4 reports the AUC measurement of OffPairC. The results in the two figures also show that the larger the training size is or privacy parameter ϵ is, the higher the AUC measurement value is, which means that the proposed algorithm is less perturbed and more accurate. These experimental results verify that the proposed online differential private algorithms can achieve good utility while guarantee strong privacy protection when they are applied to the AUC maximization task.



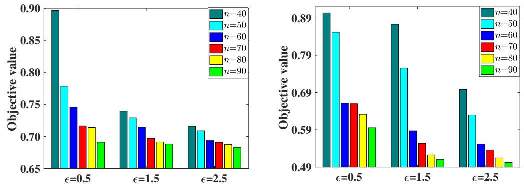
(a) Hepatitis (b) Cancer

Figure 1: The objective value of OnPairStrC for AUC maximization.



(a) Diabetes (b) Diabetic Retinopathy

Figure 2: The objective value of OnPairC for AUC maximization.



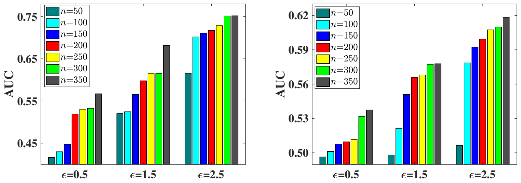
(a) Hepatitis (b) Cancer

Figure 3: The objective value of OffPairStrC for AUC maximization.

Experiments for Metric Learning

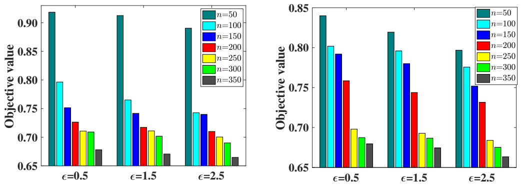
Next, we evaluate the performance of the proposed differentially private pairwise learning algorithms for the metric learning task (see Example 1 for the problem formulation). We add additional Frobenius norm $\frac{\lambda}{2}\|W\|_F^2$ to the loss function for the strongly convex case, where $\lambda = 10^{-3}$. Similar to the experiments for AUC maximization, we evaluate the effect of the privacy parameter ϵ and the training size n. Due to the space limit, in this section, we only report the experimental results for general convex pairwise learning algorithms, i.e., OnPairC and OffPairC.

In these experiments, the value of δ is fixed as $\frac{1}{n}$, and we consider three cases where the parameter ϵ is set to be 0.5, 1.5 and 2.5, respectively. We first calculate the objective value of OnPairC when the training size varies from 50 to 350, and the results on the Diabetes and Diabetic Retinopathy datasets are shown in Figure 5. As for the offline algorithm OffPairC, we report the classification accuracy in Figure 6. As we can see, the derived experimental results are similar to that for AUC maximization. The proposed algorithms perform competitively with the baseline when we vary the values of n and ϵ.



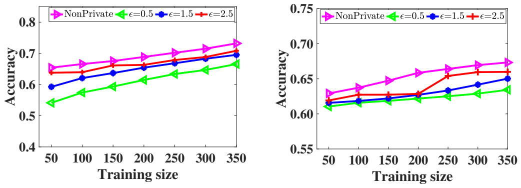
(a) Diabetes (b) Diabetic Retinopathy

Figure 4: The AUC measurement of OffPairC.



(a) Diabetes (b) Diabetic Retinopathy

Figure 5: The objective value of OnPairC for metric learning task under different training sizes.



(a) Diabetes (b) Diabetic Retinopathy

Figure 6: The classification accuracy of OffPairC for metric learning task under different training sizes.

Conclusion

In this paper, we consider the pairwise learning problems in both online and offline settings. For the online setting, we first propose an (ϵ, δ)-DP algorithm (called OnPairStrC) for the strongly convex loss functions, and then extend this algorithm to general convex loss functions by proposing another differentially private algorithm (called OnPairC). For the offline setting, we also propose two differentially private algorithms (called OffPairStrC and OffPairC) for strongly convex loss functions and general convex loss functions, respectively, and then give their regret upper bounds. The experimental results on real-world datasets not only confirm our theoretical analysis but also demonstrate the effectiveness of the proposed algorithms in real-world applications.

Acknowledgment

This work is supported in part by the US National Science Foundation under grants IIS-1910492, CCF-1716400, IIS-1924928, IIS-1938167 and OAC-1934600. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

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 (FOCS)*.

Bun, M., and Steinke, T. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *TCC*, 635–658. Springer.

Cao, Q.; Guo, Z.-C.; and Ying, Y. 2016. Generalization bounds for metric and similarity learning. *Machine Learning*.

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

Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*.

Clémençon, S.; Lugosi, G.; Vayatis, N.; et al. 2008. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*.

Dua, D., and Graff, C. 2017. UCI machine learning repository.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*. Springer.

Dwork, C.; Rothblum, G. N.; and Vadhan, S. 2010. Boosting and differential privacy. In *FOCS*, 51–60.

Fredrikson, M.; Jha, S.; and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*.

Huai, M.; Miao, C.; Li, Y.; Suo, Q.; Su, L.; and Zhang, A. 2018a. Metric learning from probabilistic labels. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 1541–1550.

Huai, M.; Miao, C.; Suo, Q.; Li, Y.; Gao, J.; and Zhang, A. 2018b. Uncorrelated patient similarity learning. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 270–278. SIAM.

Huai, M.; Xue, H.; Miao, C.; Yao, L.; Su, L.; Chen, C.; and Zhang, A. 2019. Deep metric learning: the generalization analysis and an adaptive algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2535–2541. AAAI Press.

Jain, P.; Kothari, P.; and Thakurta, A. 2012. Differentially private online learning. In *Conference on Learning Theory*, 24–1.

Jin, R.; Wang, S.; and Zhou, Y. 2009. Regularized distance metric learning: Theory and algorithm. In *NIPS*, 862–870.

Kar, P.; Sriperumbudur, B.; Jain, P.; and Karnick, H. 2013. On the generalization ability of online learning algorithms for pairwise loss functions. In *International Conference on Machine Learning*, 441–449.

Kulis, B., and Bartlett, P. L. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*.

Lee, J., and Kifer, D. 2018. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Natole, M.; Ying, Y.; and Lyu, S. 2018. Stochastic proximal algorithms for auc maximization. In *International Conference on Machine Learning*.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 3–18. IEEE.

Sun, J.; Wang, F.; Hu, J.; and Edabollahi, S. 2012. Supervised patient similarity measure of heterogeneous patient records. *ACM SIGKDD Explorations Newsletter* 14(1):16–24.

Suo, Q.; Zhong, W.; Ma, F.; Ye, Y.; Huai, M.; and Zhang, A. 2018. Multi-task sparse metric learning for monitoring patient similarity progression. In *2018 IEEE International Conference on Data Mining (ICDM)*, 477–486. IEEE.

Tang, J., and Wang, K. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proc. of the 24th SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Thakurta, A. G., and Smith, A. 2013. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, 2733–2741.

Wang, D., and Xu, J. 2019. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. *Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA, January 27-February 1, 2019*.

Wang, D.; Chen, C.; and Xu, J. 2019. Differentially private empirical risk minimization with non-convex loss functions. In Chaudhuri, K., and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6526–6535. Long Beach, California, USA: PMLR.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In *NIPS*.

Zhao, P.; Hoi, S. C.; Jin, R.; and Yang, T. 2011. Online auc maximization. In *ICML*, 233–240.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th International Conference on Machine Learning (ICML-03)*, 928–936.