Differentially Private Pairwise Learning Revisited

Zhiyu Xue[1*], **Shaoyang Yang**[2], **Mengdi Huai**[3] and **Di Wang**[4]
[1]University of Electronic Science and Technology of China
[2]Harbin Institute of Technology
[3]University of Virginia
[4]King Abdullah University of Science and Technology
di.wang@kaust.edu.sa

Abstract

Instead of learning with pointwise loss functions, learning with pairwise loss functions (pairwise learning) has received much attention recently as it is more capable of modeling the relative relationship between pairs of samples. However, most of the existing algorithms for pairwise learning fail to take into consideration the privacy issue in their design. To address this issue, previous work studied pairwise learning in the Differential Privacy (DP) model. However, their utilities (population errors) are far from optimal. To address the sub-optimal utility issue, in this paper, we proposed new pure or approximate DP algorithms for pairwise learning. Specifically, under the assumption that the loss functions are Lipschitz, our algorithms could achieve the optimal expected population risk for both strongly convex and general convex cases. We also conduct extensive experiments on real-world datasets to evaluate the proposed algorithms, experimental results support our theoretical analysis and show the priority of our algorithms.

1 Introduction

As an important family of learning problems, *pairwise learning* has drawn much attention recently. Since pairwise learning involves loss functions 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 functions only take individual samples as the input. In practice, many learning tasks can be categorized as pairwise learning problems. For instance, metric learning [Huai *et al.*, 2019] 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] and ranking [Tang and Wang, 2018], can also be categorized as pairwise learning.

Although the importance of pairwise learning has been recognized in many real-world applications, there is still a privacy issue among the current learning algorithms. 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. While DP pointwise learning has been extensively studied in the last decade, starting from [Chaudhuri and Monteleoni, 2009; Wang and Xu, 2019a; Wang *et al.*, 2017; Wang *et al.*, 2019; Wang *et al.*, 2020; Wang and Xu, 2019b; Wang and Xu, 2021; Bassily *et al.*, 2014; Bassily *et al.*, 2019; Bassily *et al.*, 2019; Feldman *et al.*, 2020]. DP pairwise learning is still not well understood. [Shang *et al.*, 2014] considered the DP for rank aggregation which combines multiple ranked lists into a single rank, their problem cannot be generalized to all pairwise loss functions. [Li *et al.*, 2020] proposed differential pairwise privacy for secure metric learning but utility (generalization) analysis is not given. Recently, [Huai *et al.*, 2020] first studied the problem under both of the online and offline settings, and provided some preliminary theoretical results, which is extended by [Yang *et al.*, 2021] to the non-smooth case. However, the problem has not been completely understood, yet. As we can see from Table 1, there is still a huge gap between their upper bounds of the population error and their corresponding lower bounds in both of the strongly convex and general convex cases, which means that their utilities are far from optimal. Motivated by this, our question is,

For the problem of differentially private pairwise learning, can we find private estimators whose population errors match their corresponding lower bounds, for strongly convex and general convex loss cases, in (ϵ, δ)/ϵ-DP model?

Here we provide the affirmative answer of the previous question, and we summarize our theoretical results in Table 1. In details, the contributions of this paper can be summarized as follows:

- Firstly, we consider the pairwise learning problem with Lipschitz, smooth and strongly convex loss functions. We propose an algorithm, which is based on the stability of the Projected Gradient Descent method, and show that its output could achieve an expected population error of $O(\frac{1}{n} + \frac{d \log \frac{1}{\delta}}{n^2\epsilon^2})$ and $O(\frac{1}{n} + \frac{d^2}{n^2\epsilon^2})$ (if we omit other terms) for (ϵ, δ)-DP and ϵ-DP, respectively, where n is the sample size and d is the dimensionality of the under-

*The first two authors contributed equally to this paper. Part of the work was done when Zhiyu Xue and Shaoyang Yang were research interns at KAUST.

	Method	(ϵ, δ)-DP		ϵ-DP	
		Upper Bound	Lower Bound	Upper Bound	Lower Bound
Strongly Convex	[Huai *et al.*, 2020]	$O(\frac{\sqrt{d}}{\sqrt{n\epsilon}})$	$\Omega(\frac{1}{n} + \frac{d}{n^2\epsilon^2})$	–	$\Omega(\frac{1}{n} + \frac{d^2}{n^2\epsilon^2})$
	This Paper	$O(\frac{1}{n} + \frac{d}{n^2\epsilon^2})$		$O(\frac{1}{n} + \frac{d^2}{n^2\epsilon^2})$	
Convex	[Huai *et al.*, 2020; Yang *et al.*, 2021]	$O(\frac{\sqrt{d}}{\sqrt{n\epsilon}})$	$\Omega(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\epsilon})$	–	$\Omega(\frac{1}{\sqrt{n}} + \frac{d}{n\epsilon})$
	This Paper	$O(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\epsilon})$		$O(\frac{1}{\sqrt{n}} + \frac{d}{n\epsilon})$	

Table 1: A summary of previous results and contributions of this paper, here we assume the loss functions are Lipschitz and Lipschitz smooth. All the bounds are for population error and all omit and other factors (such as the diameter of the constraint set). The low bounds in [Bassily *et al.*, 2019] are for pointwise loss functions, since pointwise loss is a special case of pairwise loss, thus these lower bounds still hold for pairwise loss case.

lying space. As we can see from Table 1, these bounds match their corresponding lower bounds, which means they are optimal.

- Then we study the problem with general Lipschitz and smooth convex loss functions. Unlike the strongly convex case, direct using our previous idea of proof to general convex case can only achieve a sub-optimal population error. To overcome the challenge, motivated by [Feldman *et al.*, 2020] and our previous idea, we propose an algorithm whose output could achieve an expected population error of $O(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\epsilon})$ and $O(\frac{1}{\sqrt{n}} + \frac{d}{n\epsilon})$ for (ϵ, δ)-DP and ϵ-DP, respectively. And these upper bounds are optimal.
- Finally, we conduct comprehensive experiments on metric learning and AUC maximization, with or without ℓ_2-norm regularization. Experimental results support our theoretical results and also show the priority of our algorithms compared with the previous ones.

Due to the space limit, additional definitions, related work, all the proofs are included in the full version of the paper.

2 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 $\mathcal{A}$, we have $Pr(\mathcal{A}(D) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(D') \in S) + \delta$. When $\delta = 0$, $\mathcal{A}$ is ϵ-differentially private.*

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 data samples are drawn i.i.d from an unknown underlying distribution $\mathcal{P}$ on $\mathcal{D}$, we also have the population risk, which is

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

Similar to the definition of DP pointwise learning [Bassily *et al.*, 2014], we can define DP pairwise learning as follows.

Definition 2. *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) $\{y_i\}_{i=1}^n \subset [-1, 1]^n$. Differentially private (DP) pairwise learning is to find a private estimator $w_{priv} \in \mathbb{R}^d$ 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* **population error** *$Err_{\mathcal{P}}(w) = L_{\mathcal{P}}(w) - \min_{w \in \mathcal{C}} L_{\mathcal{P}}(w)$.*

In the experiments section we will conduct experiments on metric learning and AUC maximization, with or without ℓ_2-norm regularization, for strongly convex or general convex case. Next, we will give a brief review on these two problems.
Example 1: Metric Learning [Cao *et al.*, 2016] The goal here is to learn a Mahalanobios metric $M_W^2(x, x') = (x - x')^T W (x - x')$ using loss function $\ell(W; z, z') = \phi(yy'(1 - M_W^2(x, x'))$, where $y, y' \in \{-1, +1\}$ and $\phi(x)$ is the logistic function *i.e.,* $\phi(x) = \log(1 + e^{-x})$. The constraint set $\mathcal{C}$ is $\mathcal{C} = \{W : W \in \mathbb{S}^d, \|W\|_F \leq 1\}$, where $\mathbb{S}^d$ is the set of $d \times d$ positive symmetric matrices.
Example 2: AUC Maximization [Zhao *et al.*, 2011] 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 $\ell(w; z, z') = \phi((y - y')h(w; x, x'))$ and $h(w; x, x') = w^T(x - x')$, where $y, y' \in \{-1, +1\}$.

3 Strongly Convex Case

Assumption 1: We assume the loss function $\ell(\cdot; z, z')$ is G-Lipschitz, L-smooth and α-strongly convex.

The idea of our algorithm is motivated by the ℓ_2-norm sensitivity of the Projected Gradient Descent (PGD) method for the empirical risk function. For PGD method, its ℓ_2-norm sensitivity corresponds to its stability, which has been studied in [Hardt *et al.*, 2016] for pointwise loss functions. Motivated by this, we generalize to pairwise loss functions. Based on its sensitivity and the Gaussian mechanism, we have Algorithm 1. The guarantee of DP is mainly based on the following lemma:

Algorithm 1 DP Gradient Descent-SC (DPGDSC)

Input: $D = \{z_i\}_{i=1}^n \subset \mathbb{R}^d$, privacy parameters ϵ, δ, empirical risk $L(w; D)$, initial parameter w_0, step size $\eta \leq \frac{2}{L+\alpha}$ and number of iterations T (will be specified later).

1: **for** $t = 1, 2, \cdots, T$ **do**
2: Let $w_t = \Pi_{\mathcal{C}}(w_{t-1} - \eta \nabla L(w; D))$, where $\Pi_{\mathcal{C}}$ is the projection onto the set $\mathcal{C}$.
3: **end for**
4: When $\delta > 0$, return $\tilde{w}_T = w_T + \zeta$, where $\zeta \sim \mathcal{N}(0, \sigma^2 I_d)$ and $\sigma = \frac{8\sqrt{2\ln(1.25/\delta)}G}{\alpha n \epsilon}$.
5: When $\delta = 0$, return $\tilde{w}_T = w_T + \zeta$, where $\zeta = (\zeta_1, \cdots, \zeta_d)$ with $\zeta_i \sim \text{Lap}(\lambda)$ and $\lambda = \frac{8G\sqrt{d}}{\alpha n \epsilon}$.

Lemma 1. *For any $D \sim D'$, if we denote $w_t', t \in [T]$ as the parameters which correspond to w_t in Algorithm 1 performed on D', then under Assumption 1, with $\eta \leq \frac{2}{L+\alpha}$, we have for all $t \in [T]$,*

$$\|w_t - w_t'\|_2 \leq \frac{8G}{\alpha n}. \tag{3}$$

Theorem 1. *Under Assumption 1, when the step size $\eta \leq \frac{2}{L+\alpha}$, Algorithm 1 is (ϵ, δ)-DP when $\delta > 0$ and ϵ-DP otherwise. Moreover, if we let $T = \tilde{O}(\frac{L}{\alpha}\log n)$, then when $\delta > 0$, we have*

$$\mathbb{E}_{\mathcal{A},D} Err_{\mathcal{P}}(\tilde{w}_T) \leq O(\frac{\|\mathcal{C}\|_2^2 L G^2 d \log 1/\delta}{\alpha^2 n^2 \epsilon^2} + \frac{G^2}{\alpha n}).$$

When $\delta = 0$, we have

$$\mathbb{E}_{\mathcal{A},D} Err_{\mathcal{P}}(\tilde{w}_T) \leq O(\frac{\|\mathcal{C}\|_2^2 L G^2 d^2}{\alpha^2 n^2 \epsilon^2} + \frac{G^2}{\alpha n}).$$

Where $\|\mathcal{C}\|_2$ is the diameter of the set $\mathcal{C}$ and $\tilde{O}$ omits other logarithmic factors, $\mathbb{E}_{\mathcal{A},D}$ means that the expectation takes over the randomness of the algorithm $\mathcal{A}$ and the data distribution $D \sim \mathcal{P}^n$.

Remark 1. *For pointwise loss functions, [Zhang* et al.*, 2017] provided an output perturbation method based on the ℓ_2-norm sensitivity of the PGD method. Although the ideas of these two algorithms are similar, there are still several differences on the utility guarantees. Firstly, [Zhang* et al.*, 2017] only showed that its output could achieve the optimal rate for optimality gap in the strongly convex case. However, as [Bassily* et al.*, 2019] said, optimal optimality gap of an estimator cannot guarantee its population error is also optimal. In this paper, we propose a new approach to show that our output achieves the optimal rate for the population error, which has not been studied previously. And this approach could might be used to other problems. Second, in the general convex case, as [Zhang* et al.*, 2017] said, their algorithm could only achieve a sub-optimal rate, even for the optimality gap. While in the later section we will use the idea of our approach to design an algorithm whose output could achieve the optimal rate for population error (see Section 4 for details).*

For poinwise loss case, there are mainly three approaches on showing the population errors for a given estimator w_{priv}. The first approach is to directly transfer the optimality gap to population error via some existing lemmas, such as [Bassily *et al.*, 2014; Chan *et al.*, 2011]. However, as [Bassily *et al.*, 2014] mentioned, this approach could only achieve a sub-optimal rate, see Section F of Appendix in [Bassily *et al.*, 2014] for details. The second approach is based on the online-to-batch method, which has been used in [Huai *et al.*, 2020] for pairwise loss. However, as we said previously, this approach could also only achieve a sub-optimal rate of population error. The third type of approaches is proposed by [Bassily *et al.*, 2019] recently, which is based on the uniform stability of the Differentially Private Batch SGD method. However, [Bassily *et al.*, 2019] only studied the case where the loss function is pointwise and general convex, it is unknown whether their algorithm can be extended to the pairwise loss functions or strongly convex loss functions. Our new method could be seen as an extension of the above third method. Specifically, for the output, its population error can be decomposed into the sum of its generalization error and its optimality gap [Shen *et al.*, 2020; Yang *et al.*, 2021]. Motivated by this, we bound the the optimality gap of the output via the stability of the algorithm, *i.e.,* the the ℓ_2-norm sensitivity of the PGD method.

4 General Convex Case

Motivated by the idea in the previous section, one question is whether we can generalize it to the general convex case.

Assumption 2: For any pair $z, z' \in \mathcal{D}$, we assume the loss function $\ell(\cdot; z, z')$ is convex, G-Lipschitz, and L-smooth.

The most direct problem is that whether we can use the same idea of Algorithm 1, *i.e.,* perturbing the output of PGD method. We show that it is possible. However, the population error of our output is only sub-optimal in the general convex case, which is quite different compared with the strongly convex case.

In the next, we propose a simple method and show that for (ϵ, δ)-DP, instead of perturbing the output of the PGD method, perturbing the gradient by Gaussian noise in each iteration of PGD method could directly achieve the optimal rate of population error. It is notable that although many previous paper also studied Algorithm 2 [Bassily *et al.*, 2014], most of them only considered the optimality gap. However, in this paper we focus on the optimality of population error.

Algorithm 2 DP Gradient Descent (DPGDC2)

Input: $D = \{z_i\}_{i=1}^n \subset \mathbb{R}^d$, privacy parameters $\epsilon, \delta > 0$; empirical risk $L(w; D)$, initial parameter w_0, step size $\eta \leq \frac{2}{L}$ and number of iterations T.

1: **for** $t = 1, 2, \cdots, T$ **do**
2: Let $w_t = \Pi_{\mathcal{C}}(w_{t-1} - \eta(\nabla L(w; D) + \zeta_i))$, where $\zeta_i \sim \mathcal{N}(0, \sigma^2 I_d)$ with $\sigma = \frac{4G\sqrt{1.25T\log 1/\delta}}{n\epsilon}$
3: **end for**
4: Return $\bar{w}_T = \sum_{i=0}^T \frac{w_0 + \cdots + w_T}{T+1}$

Theorem 2. *Under Assumption 2, Algorithm 2 is (ϵ, δ)-DP. Moreover, we have the following by setting $T = \min\{n, \frac{n^2\epsilon^2}{d\log 1/\delta}\}$ and $\eta = \frac{G}{\|\mathcal{C}\|_2\sqrt{T}}$ if $L \leq \frac{\|\mathcal{C}\|_2}{2G}\min\{n, \frac{n\epsilon}{\sqrt{d\log 1/\delta}}\}$*

$$\mathbb{E}_{\mathcal{A},D}Err_{\mathcal{P}}(\bar{w}_T) \leq O(G\|\mathcal{C}\|_2(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log 1/\delta}}{n\epsilon})).$$

While Algorithm 2 is succinct, there are still many issues. Firstly, Theorem 2 only holds for (ϵ, δ)-DP model, it is unknown whether we can extend to ϵ-DP model. Secondly, we can see that in Algorithm 2 the privacy budget is evenly split across iterations. However, as we know when the iteration number increases, our estimator will be closed to the optimal one and the gradients start to decrease and need to be measured more accurately in order for the optimization to continue making progress. This means that an adaptive privacy budget allocation may has preferable practical performance to a fixed allocation, as long as the total privacy cost is the same.

To address the above two issues, we propose a new method which is based on [Feldman *et al.*, 2020]. The idea is that, for poinwise loss functions in the non-private case, compared with the PGD method, recently some work such as [Hazan and Kale, 2014; Feldman *et al.*, 2020] showed that a variant of PGD, which is called the Epoch PGD method, could achieve an improved bound of generalization error. The basic idea of Epoch PGD is that, we first divide the whole dataset into several disjoint subsets; in each epoch, we run the PGD method for several iterations on one of these subsets; then we take the current parameter as the initial parameter of the next epoch. Motivated by this, we propose a DP version of the Epoch PGD method (for convenience here we assume $n = 2^k$ for some positive integer k). We have the following theoretical guarantees.

Theorem 3. *Under Assumption 2, when the step size $\eta \leq \frac{2}{L}$, Algorithm 3 is (ϵ, δ)-DP when $\delta > 0$ and ϵ-DP otherwise. Moreover, when $\delta > 0$, we have the following result by setting $\eta = \frac{\|\mathcal{C}\|_2}{G}\min\{\frac{4}{\sqrt{n}}, \frac{\epsilon}{\sqrt{d\log 1/\delta}}\}$*

$$\mathbb{E}_{\mathcal{A},D}Err_{\mathcal{P}}(\tilde{w}_T) \leq O(G\|\mathcal{C}\|_2(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log 1/\delta}}{n\epsilon})).$$

When $\delta = 0$, setting $\eta = \frac{\|\mathcal{C}\|_2}{G}\min\{\frac{4}{\sqrt{n}}, \frac{\epsilon}{d}\}$ we have

$$\mathbb{E}_{\mathcal{A},D}Err_{\mathcal{P}}(\tilde{w}_T) \leq O(G\|\mathcal{C}\|_2(\frac{1}{\sqrt{n}} + \frac{d}{n\epsilon})).$$

Remark 2. *Compared with Algorithm 2, Algorithm 3 could achieve the optimal rate for both (ϵ, δ)-DP and ϵ-DP models. Moreover, since the stepsize in each epoch is varied and the magnitude of the noise depends on the stepsize, the noise we added in each epoch is different and adaptive. More specifically, as we can see from Theorem 3, when the sample size is large enough, the stepsize η_i will be very small and it will be decayed to $4^{-i}\eta$ in the i-th epoch, this means that it will be closed to 0, and thus the noise we add will be closed to 0 when the iteration number increases. This means that the practical performance of Algorithm 3 will be better than Algorithm 2, we will verify this conclusion in the experimental part.*

Algorithm 3 DP Epoch Gradient Descent (DPEGD)

Input: $D = \{z_i\}_{i=1}^n \subset \mathbb{R}^d$, privacy parameters ϵ, δ, empirical risk $L(w; D)$, initial parameter w_0, step size $\eta \leq \frac{2}{L}$.

1: Let $k = \log_2 n$, we divide the dataset D into k disjoint subsets $\{D_1, \cdots, D_k\}$, where each D_i has $n_i = 2^{-i}n$ samples for $i < k$, and D_k contains all the left data samples.
2: **for** $i = 1, 2, \cdots, k$ **do**
3: Let $\eta_i = 4^{-i}\eta$.
4: Run the PGD method for $L(\cdot; D_i)$ on the constraint set $\mathcal{C}$ and we take w_{i-1} as the initial parameter. Specifically, we set the fixed stepsize as η_i and the iteration number as n_i. Let $\bar{w}_i$ be the average parameter after n_i iterations.
5: When $\delta > 0$, let $w_i = \bar{w}_i + \zeta_i$, where $\zeta_i \sim \mathcal{N}(0, \sigma^2 I_d)$ and $\sigma = \frac{4\sqrt{2\ln(1.25/\delta)}G\eta_i}{\epsilon}$
6: When $\delta = 0$, let $w_i = \bar{w}_i + \zeta_i$, where $\zeta_i = (\zeta_1, \cdots, \zeta_d)$ with each $\zeta_j \sim \text{Lap}(\lambda)$ and $\lambda = \frac{4G\eta_i\sqrt{d}}{\epsilon}$
7: **end for**
8: Return w_k

5 Experiments

Datasets. We use two real-world datasets that are widely adopted in pairwise learning tasks. These datasets are the Diabetes dataset and the Diabetic Retinopathy dataset, which have also been used in [Huai *et al.*, 2020].

Performance measures. To evaluate the performance of the proposed algorithms, we use the following measures:

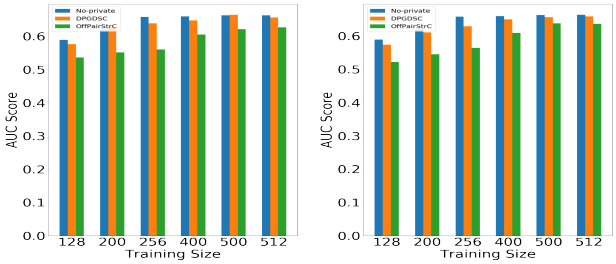
- Classification Accuracy: For metric learning task, we calculate the classification accuracy that is defined as the percentage of the correctly classified samples in the test set. The less the classification accuracy, the worse the performance of the proposed algorithm. In this paper, the KNN classifier is adopted to assign labels to the test samples. For the KNN classifier, we set K to be 3.
- AUC Score: For AUC maximization task, we report the AUC score [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.

Baseline methods. As we mentioned before, [Huai *et al.*, 2020] is the only work on DP pairwise learning, thus we use OffPairStrC and OffPairC proposed in [Huai *et al.*, 2020] for strongly convex and convex case as our baselines for private algorithms, respectively. We will also follow [Huai *et al.*, 2020] and use variants of OffPairStrC and OffPairC, which do not add any noise, as non-private baseline methods. In these experiments, we will choose different ϵ. And for (ϵ, δ)-DP model, we will set $\delta = \frac{1}{n}$.

Experimental settings. In this paper we studied both of the strongly convex and general convex cases. To conduct experiments for strongly convex case, we add an additional Frobenius norm or ℓ_2-norm regularization term with some $\lambda > 0$ to the original problem of metric learning and AUC maximization respectively to make the loss be strongly convex. We set $\lambda = 10^{-3}$ for AUC maximization and $\lambda = 10^{-2}$ for metric learning.

Metric Learning. In Table 2 we perform the results for different training sample size, with fixed privacy budget $\epsilon = 1$. And in Table 3 we show the results for different privacy budget, with fixed training sample size $n = 512$. Compared with previous methods, our algorithms show better performance under all the four different settings:

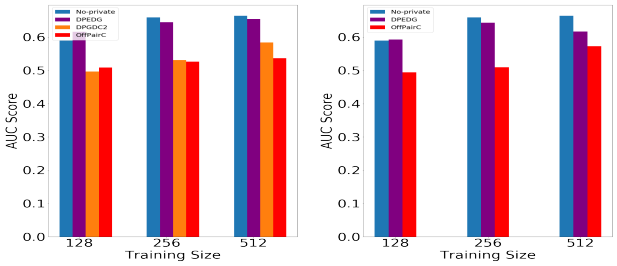
- In the strongly convex case and $\delta > 0$, DPGDSC (Algorithm 1) performs better than OffPairStrC and the difference of accuracy between them increases as the training size increases, and it will be closed to the non-private case. Furthermore, if we fix the training size and change the parameter ϵ, we can see from Table 3 that DPGDSC maintains its advantage over OffPairStrC.
- When the loss function is convex and $\delta > 0$, DPGDC2 (Algorithm 2) shows an improvement in comparison with OffPairC. Especially, it has significant improvement on the Diabetes dataset. In addition, DPEGD (Algorithm 3) has better performance than OffPairC and DPGDC2 on both datasets. Moreover, from Table 3 we can see under different ϵ, DPEGD outperforms other methods.
- In the strongly convex case in the ϵ-DP model, although the improvement is limited, we can still see that our new algorithm is slightly better than the best known method. Moreover, as shown in Table 3, except for some cases, most of the results show that DPGDSC has a better performance than OffPairStrC.
- Finally, we can see that, when the loss function is convex and in the ϵ-DP model, DPEGD outperforms OffPairC under different ϵ or different training sample size.



(a) $\delta = \frac{1}{n}$ (b) $\delta = 0$

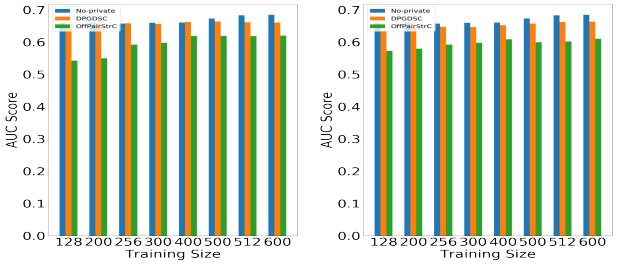
Figure 1: AUC maximization: Results for different training size in strongly convex case on Diabetes dataset, where $\epsilon = 0.8$.

AUC Maximization. For AUC maximization, Table 4 shows the results on Diabetes and Diabetic Retinopathy datasets for different ϵ with fixed $n = 256$. Figure 1, 2, 3 and 4 shows the results for different sample size in strongly convex or general convex case, under (ϵ, δ) or ϵ-DP model respectively, with fixed $\epsilon = 0.8$. From these results, we can get almost the same conclusions as in the metric learning case. Moreover, from Figure 1(b) and 3(b), we can see when the loss function is strongly convex, the performance of DPGDSC is much better than OffPairStrC, while the difference of accuracy between these two methods is quite small in the metric learning task.



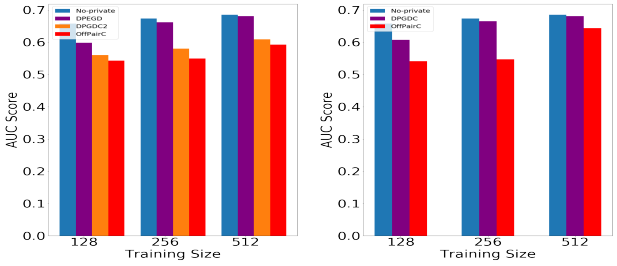
(a) $\delta = \frac{1}{n}$ (b) $\delta = 0$

Figure 2: AUC maximization: Results for different training size in general convex case on Diabetes dataset, where $\epsilon = 0.8$.



(a) $\delta = \frac{1}{n}$ (b) $\delta = 0$

Figure 3: AUC maximization: Results for different training size in strongly convex case on Diabetic Retinopathy dataset, where $\epsilon = 0.8$.



(a) $\delta = \frac{1}{n}$ (b) $\delta = 0$

Figure 4: AUC maximization: Results for different training size in general convex case on Diabetic Retinopathy dataset, where $\epsilon = 0.8$.

Loss function	Algorithm	Training size					
		Diabetes			Diabetic Retinopathy		
		128	256	512	128	256	512
Strongly convex $\delta \neq 0$	Non-private	71.40%	72.39%	72.88%	62.82%	63.84%	65.01%
	OffPairStrC	63.69%	64.55%	64.63%	60.72%	62.14%	63.59%
	DPGDSC	**64.03%**	**64.68%**	**65.85%**	**59.72%**	**62.82%**	**65.13%**
General convex $\delta \neq 0$	Non-private	71.73%	72.52%	72.97%	61.57%	63.86%	65.03%
	OffPairC	64.20%	64.64%	65.87%	60.94%	62.85%	63.29%
	DPGDC2	**71.30%**	**71.91%**	**72.46%**	**62.32%**	**63.09%**	**64.35%**
	DPEGD	**71.29%**	**72.21%**	**72.84%**	**62.95%**	**65.21%**	**66.36%**
Strongly convex δ=0	Non-private	71.71%	71.99%	72.56%	62.39%	63.13%	65.49%
	OffPairStrC	64.37%	65.64%	66.77%	59.32%	61.00%	61.78%
	DPGDSC	**64.51%**	**65.28%**	**67.16%**	**59.48%**	**61.07%**	**62.01%**
General convex δ=0	Non-private	71.80%	72.47%	72.80%	61.84%	63.42%	65.31%
	OffPairC	64.97%	65.58%	67.28%	59.55%	60.70%	61.77%
	DPEGD	**70.37%**	**71.16%**	**71.24%**	**63.41%**	**64.51%**	**66.54%**

Table 2: Metric learning: Experimental results on Diabetes and Diabetic Retinopathy dataset for different training sizes with fixed $\epsilon = 1$.

Loss function	Dataset	Algorithm	ϵ			ϵ		
			0.2	0.5	0.8	1.0	1.5	2.0
Strongly convex $\delta \neq 0$	Diabetes	OffPairStrC	63.49%	63.50%	63.93%	63.44%	63.53%	64.26%
		DPGDSC	**64.18%**	**64.92%**	**65.72%**	**63.91%**	**64.01%**	**64.29%**
	Diabetic Retinopathy	OffPairStrC	60.30%	60.40%	60.47%	63.44%	63.53%	64.26%
		DPGDSC	**60.63%**	**61.81%**	**62.57%**	**63.91%**	**64.01%**	**64.29%**
General convex $\delta \neq 0$	Diabetes	OffPairC	63.59%	63.63%	63.97%	63.71%	63.96%	65.07%
		DPGDC2	**71.72%**	**70.61%**	**72.11%**	**71.05%**	**70.83%**	**71.36%**
		DPEGD	**71.46%**	**71.49%**	**71.66%**	**71.32%**	**71.50%**	**71.45%**
	Diabetic Retinopathy	OffPairC	60.21%	60.29%	60.71%	63.71%	63.96%	65.07%
		DPGDC2	**61.27%**	**61.79%**	**60.87%**	**71.05%**	**70.83%**	**71.36%**
		DPEGD	**62.58%**	**62.84%**	**62.89%**	**71.32%**	**71.50%**	**71.45%**
Strongly convex δ=0	Diabetes	OffPairStrC	64.28%	64.49%	64.53%	64.38%	64.40%	64.84%
		DPGDSC	**64.45%**	**64.84%**	**64.84%**	**64.63%**	**64.79%**	**64.81%**
	Diabetic Retinopathy	OffPairStrC	59.54%	59.57%	59.60%	64.38%	64.40%	64.84%
		DPGDSC	**59.60%**	**59.70%**	**59.50%**	**64.63%**	**64.79%**	**64.81%**
General convex δ=0	Diabetes	OffPairC	64.34%	64.38%	64.49%	64.09%	64.29%	64.30%
		DPEGD	**70.28%**	**70.49%**	**70.51%**	**70.48%**	**70.59%**	**70.82%**
	Diabetic Retinopathy	OffPairC	59.54%	59.59%	59.80%	64.09%	64.29%	64.30%
		DPEGD	**62.87%**	**62.84%**	**62.87%**	**70.48%**	**70.59%**	**70.82%**

Table 3: Metric learning: Experimental results on Diabetes and Diabetic Retinopathy dataset for different ϵ with fixed $n = 128$.

Loss function	Dataset	Algorithm	ϵ		ϵ	
			0.5	0.8	1.0	2.0
Strongly convex $\delta \neq 0$	Diabetes	OffPairStrC	53.71%	56.05%	59.52%	64.93%
		DPGDSC	**63.26%**	**63.92%**	**64.46%**	**65.51%**
	Diabetic Retinopathy	OffPairStrC	56.33%	59.27%	62.92%	67.01%
		DPGDSC	**65.65%**	**66.30%**	**67.23%**	**67.04%**
General convex $\delta \neq 0$	Diabetes	OffPairC	52.01%	52.62%	54.51%	57.44%
		DPGDC2	**52.94%**	**53.09%**	**54.61%**	**59.96%**
		DPEGD	**64.52%**	**64.47%**	**64.41%**	**64.37%**
	Diabetic Retinopathy	OffPairC	50.08%	52.90%	54.27%	62.92%
		DPGDC2	**54.37%**	**58.06%**	**60.03%**	**60.44%**
		DPEGD	**66.19%**	**66.21%**	**66.29%**	**66.09%**
Strongly convex $\delta = 0$	Diabetes	OffPairStrC	50.65%	56.45%	59.94%	64.13%
		DPGDSC	**59.16%**	**62.98%**	**62.67%**	**64.63%**
	Diabetic Retinopathy	OffPairStrC	52.24%	54.74%	57.54%	66.25%
		DPGDSC	**62.75%**	**64.56%**	**65.47%**	**66.94%**
General convex $\delta = 0$	Diabetes	OffPairC	50.25%	50.90%	52.57%	60.13%
		DPEGD	**59.16%**	**64.35%**	**64.50%**	**64.47%**
	Diabetic Retinopathy	OffPairC	52.26%	50.13%	51.43%	58.06%
		DPEGD	**66.34%**	**66.50%**	**66.04%**	**66.38%**

Table 4: AUC maximization: Experimental results on Diabetes and Diabetic Retinopathy dataset for different ϵ, where $n = 256$

References

[Bassily *et al.*, 2014] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 2014.

[Bassily *et al.*, 2019] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, pages 11282–11291, 2019.

[Cao *et al.*, 2016] Qiong Cao, Zheng-Chu Guo, and Yiming Ying. Generalization bounds for metric and similarity learning. *Machine Learning*, 2016.

[Chan *et al.*, 2011] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 2011.

[Chaudhuri and Monteleoni, 2009] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, 2009.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*. Springer, 2006.

[Feldman *et al.*, 2020] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

[Hardt *et al.*, 2016] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

[Hazan and Kale, 2014] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

[Huai *et al.*, 2019] Mengdi Huai, Hongfei Xue, Chenglin Miao, Liuyi Yao, Lu Su, Changyou Chen, and Aidong Zhang. Deep metric learning: the generalization analysis and an adaptive algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2535–2541. AAAI Press, 2019.

[Huai *et al.*, 2020] Mengdi Huai, Di Wang, Chenglin Miao, Jinhui Xu, and Aidong Zhang. Pairwise learning with differential privacy guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 694–701, 2020.

[Li *et al.*, 2020] Jing Li, Yuangang Pan, Yulei Sui, and Ivor W Tsang. Secure metric learning via differential pairwise privacy. *IEEE Transactions on Information Forensics and Security*, 2020.

[Shang *et al.*, 2014] Shang Shang, Tiance Wang, Paul Cuff, and Sanjeev Kulkarni. The application of differential privacy for rank aggregation: Privacy and accuracy. In *17th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2014.

[Shen *et al.*, 2020] Wei Shen, Zhenhuan Yang, Yiming Ying, and Xiaoming Yuan. Stability and optimization error of stochastic gradient descent for pairwise learning. *Analysis and Applications*, 18(05):887–927, 2020.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. 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*, 2018.

[Wang and Xu, 2019a] Di Wang and Jinhui Xu. 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*, 2019.

[Wang and Xu, 2019b] Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. In *International Conference on Machine Learning*, pages 6628–6637, 2019.

[Wang and Xu, 2021] Di Wang and Jinhui Xu. Escaping saddle points of empirical risk privately and scalably via dp-trust region method. In *Machine Learning and Knowledge Discovery in Databases*, pages 90–106. Springer International Publishing, 2021.

[Wang *et al.*, 2017] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[Wang *et al.*, 2019] Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535, 2019.

[Wang *et al.*, 2020] Di Wang, Hanshen Xiao, Srini Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. *arXiv preprint arXiv:2010.11082*, 2020.

[Yang *et al.*, 2021] Zhenhuan Yang, Yunwen Lei, Siwei Lyu, and Yiming Ying. Stability and differential privacy of stochastic gradient descent for pairwise learning with non-smooth loss. In *International Conference on Artificial Intelligence and Statistics*, pages 2026–2034. PMLR, 2021.

[Zhang *et al.*, 2017] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. In *IJCAI*, pages 3922–3928, 2017.

[Zhao *et al.*, 2011] Peilin Zhao, Steven CH Hoi, Rong Jin, and Tianbao Yang. Online auc maximization. In *ICML*, pages 233–240, 2011.