A Knowledge Transfer Framework for Differentially Private Sparse Learning

Lingxiao Wang, Quanquan Gu
Department of Computer Science, University of California, Los Angeles
{lingxw, qgu}@cs.ucla.edu

Abstract

We study the problem of estimating high dimensional models with underlying sparse structures while preserving the privacy of each training example. We develop a differentially private high-dimensional sparse learning framework using the idea of knowledge transfer. More specifically, we propose to distill the knowledge from a "teacher" estimator trained on a private dataset, by creating a new dataset from auxiliary features, and then train a differentially private "student" estimator using this new dataset. In addition, we establish the linear convergence rate as well as the utility guarantee for our proposed method. For sparse linear regression and sparse logistic regression, our method achieves improved utility guarantees compared with the best known results (Kifer, Smith and Thakurta 2012; Wang and Gu 2019). We further demonstrate the superiority of our framework through both synthetic and real-world data experiments.

1 Introduction

In the Big Data era, sensitive data such as genomic data and purchase history data, are ubiquitous, which necessitates learning algorithms that can protect the privacy of each individual data record. A rigorous and standard notion for privacy guarantees is differential privacy (Dwork et al. 2006). By adding random noise to the model parameters (output perturbation), some intermediate steps of the learning algorithm (gradient perturbation), or the objective function of learning algorithms (objective perturbation), differentially private algorithms ensure that the trained models can learn the statistical information of the population without leaking any information about the individuals. In the last decade, a surge of differentially private learning algorithms (Chaudhuri and Monteleoni 2009; Chaudhuri, Monteleoni, and Sarwate 2011; Kifer, Smith, and Thakurta 2012; Bassily, Smith, and Thakurta 2014; Talwar, Thakurta, and Zhang 2015; Zhang et al. 2017; Wang, Ye, and Xu 2017; Wang, Gaboardi, and Xu 2018; Jayaraman et al. 2018; Wang et al. 2019) for empirical risk minimization have been developed. However, most of these studies only consider the classical setting, where the problem dimension is fixed. In the modern high-dimensional setting where the problem dimension can increase with the number of observations, all these empirical risk minimization algorithms fail. A common and effective approach to address these issues is to assume the model has a certain structure such as sparse structure or low-rank structure. In this paper, we consider high-dimensional models with sparse structure. Given a dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ are the input vector and response of the i-th example, our goal is to estimate the underlying sparse parameter vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$, which has s^* nonzero entries, by solving the following ℓ_2-norm regularized optimization problem with the sparsity constraint

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \bar{L}_S(\boldsymbol{\theta}) := L_S(\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}\|_2^2/2 \quad \text{subject to} \quad \|\boldsymbol{\theta}\|_0 \leq s, \tag{1.1}$$

where $L_S(\boldsymbol{\theta}) := n^{-1}\sum_{i=1}^n \ell(\boldsymbol{\theta}; \mathbf{x}_i, y_i)$ is the empirical loss on the training data, $\ell(\boldsymbol{\theta}; \mathbf{x}_i, y_i)$ is the loss function defined on the training example $(\mathbf{x}_i, y_i)$, $\lambda \geq 0$ is a regularization parameter, $\|\boldsymbol{\theta}\|_0$ counts the number of nonzero entries in $\boldsymbol{\theta}$, and s controls the sparsity of $\boldsymbol{\theta}$. The reason we add an extra ℓ_2 regularizer to (1.1) is to ensure the strong convexity of the objective function without making any assumption on the data.

In order to achieve differential privacy for sparse learning, a line of research (Kifer, Smith, and Thakurta 2012; Thakurta and Smith 2013; Jain and Thakurta 2014; Talwar, Thakurta, and Zhang 2015; Wang and Gu 2019) studied differentially private learning problems in the high-dimensional setting, where the problem dimension can be larger than the number of observations. For example, (Jain and Thakurta 2014) provided a differentially private algorithm with the dimension independent utility guarantee. However, their approach only considers the case when the underlying parameter lies in a simplex. For sparse linear regression, (Kifer, Smith, and Thakurta 2012; Thakurta and Smith 2013) proposed a two-stage approach to ensure differentially privacy. In detail, they first estimate the support set of the sparse model parameter vector using some differentially private model selection algorithm, and then estimate the parameter vector with its support restricted to the estimated subset using the objective perturbation ap-

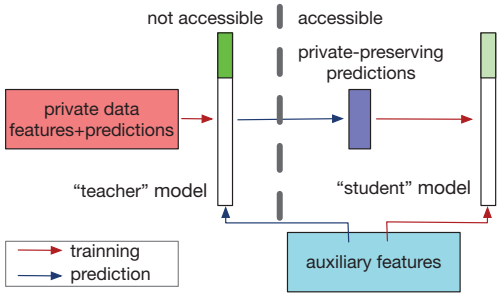


Figure 1: Illustration of the proposed framework: (1). A "teacher" estimator is trained using the private dataset; (2). A new private-preserving dataset is generated using the auxiliary features and their private predictions output by the "teacher" estimator; (3). A differentially private "student" estimator is trained using the newly generated dataset.

proach (Chaudhuri and Monteleoni 2009). Nevertheless, the support selection algorithm, like exponential mechanism, is computational inefficient or even intractable in practice. (Talwar, Thakurta, and Zhang 2015) proposed a differentially private algorithm for sparse linear regression by combining the Frank-Wolfe method (Frank and Wolfe 1956) and the exponential mechanism. Although their utility guarantee is worse than (Kifer, Smith, and Thakurta 2012; Wang and Gu 2019), it does not depend on the restricted strong convexity (RSC) and smoothness (RSS) conditions (Negahban et al. 2009). Recently, (Wang and Gu 2019) developed a differentially private iterative gradient hard thresholding (IGHT) (Jain, Tewari, and Kar 2014; Yuan, Li, and Zhang 2014) based framework for sparse learning problems by injecting Gaussian noise into the intermediate gradients. However, all the aforementioned methods either have unsatisfactory utility guarantees or are computationally inefficient. For example, the utility guarantees provided by (Kifer, Smith, and Thakurta 2012; Thakurta and Smith 2013; Wang and Gu 2019) depend on the ℓ_2-norm bound of the input vector, which can be in the order of $O(\sqrt{d})$ and grows as d increases in the worse case. While the utility guarantee of the algorithm proposed by (Talwar, Thakurta, and Zhang 2015) only depends on the ℓ_∞-norm bound of the input vector, it has a worse utility guarantee, and its convergence rate is sub-linear.

Therefore, a natural question is whether we can achieve the best of both worlds: a strong utility guarantee and high computational efficiency. To this end, we propose to make use of the idea of knowledge distillation (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015), which is a knowledge transfer technique originally introduced as a mean of model compression. The original motivation of using knowledge distillation is to use a large and complex "teacher" model to train a small "student" model, while maintaining its accuracy. For the differentially private sparse learning problem, similar idea can be applied here: we can use a non-private "teacher" model to train a differentially private "student" model, while preserving the sparse information of the "teacher" model. We notice that several knowledge transfer approaches have been recently investigated in the differentially private classification problem (Hamm, Cao, and Belkin 2016; Papernot et al. 2016; Bassily, Thakkar, and Thakurta 2018; Yoon, Jordon, and van der Schaar 2018). Nevertheless, the application of knowledge distillation to the generic differentially private high-dimensional sparse learning problem is new and has never been studied before.

In this paper, we propose a knowledge transfer framework for solving the high-dimensional sparse learning problem on a private dataset, which is illustrated in Figure 1. Our proposed algorithm is not only very efficient but also has improved utility guarantees compared with the state-of-the-art methods. More specifically, we first train a non-private "teacher" model using IGHT from the private dataset. Based on this "teacher" model, we then construct a privacy-preserving dataset using some auxiliary inputs, which are drawn from some given distributions or public datasets. Finally, by training a "student" model using IGHT again based on the newly generated dataset, we can obtain a differentially private sparse estimator. Table 1 summarizes the detailed comparisons of different methods for sparse linear regression, and we summarize the contributions of our work as follows

- Our proposed differentially private framework can be applied to any smooth loss function, which covers a broad family of sparse learning problems. In particular, we showcase the application of our framework to sparse linear regression and sparse logistic regression.
- We prove a better utility guarantee and establish a liner convergence rate for our proposed method. For example, for sparse linear regression, our method achieves $O\big(K^2 s^{*2}\sqrt{\log d}/(n\epsilon)\big)$ utility guarantee, where K is the ℓ_∞-norm bound of the input vectors, and ϵ is the privacy budget. Compared with the best known utility bound $O\big(\widetilde{K}^2 s^{*2}\log d/(n^2\epsilon^2)\big)$ (Kifer, Smith, and Thakurta 2012; Wang and Gu 2019) ($\widetilde{K}$ is the ℓ_2-norm bound of the input vectors), our utility guarantee is better than it by a factor of $O\big(\widetilde{K}^2\sqrt{\log d}/(K^2 n\epsilon)\big)$. Considering that $\widetilde{K}$ can be $\sqrt{d}$ times larger than K, the improvement factor can be as large as $O\big(d\sqrt{\log d}/(n\epsilon)\big)$. Similar improvement is achieved for sparse logistic regression.
- With the extra sparse eigenvalue condition (Bickel et al. 2009) on the private data, our method can achieve $O\big(K^2 s^{*3}\log d/(n^2\epsilon^2)\big)$ utility guarantee for sparse linear regression. It is better than the best known result (Kifer, Smith, and Thakurta 2012; Wang and Gu 2019) $O\big(\widetilde{K}^2 s^{*2}\log d/(n^2\epsilon^2)\big)$ by a factor of $O\big(\widetilde{K}^2/(K^2 s^*)\big)$, which can be as large as $O\big(d/s^*\big)$. Similar improvement is also achieved for sparse logistic regression.

Notation. For a d-dimensional vector $\mathbf{x} = [x_1, ..., x_d]^\top$, we use $\|\mathbf{x}\|_2 = (\sum_{i=1}^d |x_i|^2)^{1/2}$ to denote its ℓ_2-norm, and use $\|\mathbf{x}\|_\infty = \max_i |x_i|$ to denote its ℓ_∞-norm. We let $\mathrm{supp}(\mathbf{x})$ be the index set of nonzero entries of $\mathbf{x}$, and $\mathrm{supp}(\mathbf{x}, s)$ be the index set of the top s entries of $\mathbf{x}$ in terms of magnitude. We use $\mathcal{S}^n$ to denote the input space with n examples

Table 1: Comparison of different algorithms for sparse linear regression in the setting of (ϵ, δ)-DP. We report the utility bound achieved by the privacy-preserving mechanisms, and ignore the $\log(1/\delta)$ term. Note that $n\epsilon \gg 1$, $\mathbf{x}_i$ denotes the i-th input vector, and υ is the probability that the support selection procedure can successfully recover the true support.

Algorithm	Data Assumption	Utility	Convergence Rate	Utility Assumption
Frank-Wolfe (Talwar, Thakurta, and Zhang 2015)	$\max_{i\in[n]} \|\mathbf{x}_i\|_\infty \leq 1$	$O\Big(\frac{\log(nd)}{(n\epsilon)^{2/3}}\Big)$	Sub-linear	No
Two Stage (Kifer, Smith, and Thakurta 2012)	$\max_{i\in[n]} \|\mathbf{x}_i\|_2 \leq \widetilde{K}$	$O\Big(\frac{\widetilde{K}^2 s^{*2}\log(2/\upsilon)}{(n\epsilon)^2}\Big)$	NA	RSC/RSS
DP-IGHT (Wang and Gu 2019)	$\max_{i\in[n]} \|\mathbf{x}_i\|_2 \leq \widetilde{K}$	$O\Big(\frac{\widetilde{K}^2 s^{*2}\log d}{(n\epsilon)^2}\Big)$	Linear	RSC/RSS
DPSL-KT $\lambda > 0$	$\max_{i\in[n]} \|\mathbf{x}_i\|_\infty \leq K$	$O\Big(\frac{K^2 s^{*2}\sqrt{\log d}}{n\epsilon}\Big)$	Linear	No
DPSL-KT $\lambda = 0$	$\max_{i\in[n]} \|\mathbf{x}_i\|_\infty \leq K$ RSC/RSS	$O\Big(\frac{K^2 s^{*3}\log d}{(n\epsilon)^2}\Big)$	Linear	RSC/RSS

and $\mathcal{R}, \mathcal{R}'$ to denote the output space. Given two sequences $\{a_n\}, \{b_n\}$, if there exists a constant $0 < C < \infty$ such that $a_n \leq Cb_n$, we write $a_n = O(b_n)$, and we use $\widetilde{O}(\cdot)$ to hide the logarithmic factors. We use $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ to denote the identity matrix. Throughout the paper, we use $\ell_i(\cdot)$ as the shorthand notation for $\ell(\cdot; \mathbf{x}_i, y_i)$, and $\boldsymbol{\theta}_{\min}$ to denote the minimizer of problem (1.1).

1.1 Additional Related Work

To further enhance the privacy guarantee for training data, there has emerged a fresh line of research (Hamm, Cao, and Belkin 2016; Papernot et al. 2016; Bassily, Thakkar, and Thakurta 2018; Yoon, Jordon, and van der Schaar 2018) that studies the knowledge transfer techniques for the differentially private classification problem. More specifically, these methods propose to first train an ensemble of "teacher" models based on disjoint subsets of the private dataset, and then train a "student" model based on the private aggregation of the ensemble. However, their approaches only work for the classification task, and cannot be directly applied to general sparse learning problems. Moreover, their subsample and aggregate framework may not be suitable for the high-dimensional sparse learning problem since each "teacher" model is trained on a subset of the private dataset, which makes the "large d, small n" scenario even worse. In contrast to their sub-sample and aggregate based knowledge transfer approach, we propose to use the distillation based method (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015), which is more suitable for the high-dimensional sparse learning problem.

2 Preliminaries

In this section, we introduce some background and preliminaries about optimization and differential privacy. We first lay out the formal definitions of strongly convex and smooth functions.

Definition 2.1. A function $f : \mathbb{R}^d \to \mathbb{R}$ is λ-strongly convex, if for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$,

$$f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2) - \langle \nabla f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \geq \lambda\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2/2.$$

Definition 2.2. A function $f : \mathbb{R}^d \to \mathbb{R}$ is $\bar{\beta}$-smooth, if for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$,

$$f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2) - \langle \nabla f(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \leq \bar{\beta}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2/2.$$

Next we present the definition of sub-Gaussian distribution (Vershynin 2010).

Definition 2.3. We say $\boldsymbol{X} \in \mathbb{R}^d$ is a sub-Gaussian random vector with parameter $\alpha > 0$, if $(\mathbb{E}|\mathbf{u}^\top \boldsymbol{X}|^p)^{1/p} \leq \alpha\sqrt{p}$ for all $p \geq 1$ and all unit vector $\mathbf{u}$ with $\|\mathbf{u}\|_2 = 1$.

We also provide the definition of differential privacy.

Definition 2.4 ((Dwork et al. 2006))**.** A randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one example, and any output subset $O \subseteq \mathcal{R}$, it holds that $\mathbb{P}[\mathcal{M}(S) \in O] \leq e^\epsilon \cdot \mathbb{P}[\mathcal{M}(S') \in O] + \delta$, where $\delta \in [0, 1)$.

Now we introduce the Gaussian Mechanism (Dwork and Roth 2014) to achieve (ϵ, δ)-DP. We start with the definition of ℓ_2-sensitivity, which is used to control the variance of the noise in Gaussian mechanism.

Definition 2.5 ((Dwork and Roth 2014))**.** For two adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one example, the ℓ_2-sensitivity $\Delta_2(q)$ of a function $q : \mathcal{S}^n \to \mathbb{R}^d$ is defined as $\Delta_2(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$.

Given the ℓ_2-sensitivity, we can ensure the differential privacy using Gaussian mechanism.

Lemma 2.6. The Gaussian Mechanism $\mathcal{M} = q(S) + \mathbf{u}$, where $q : \mathcal{S}^n \to \mathbb{R}^d$ and $\mathbf{u} \sim N(0, \sigma^2\mathbf{I}_d)$, satisfies (ϵ, δ)-DP for some $\delta > 0$, if $\sigma = \sqrt{2\log(1.25/\delta)}\Delta_2(q)/\epsilon$.

The above lemma is established in (Dwork and Roth 2014). The original lemma has a constraint on $\epsilon \in (0, 1)$, which can be removed using the notion of Renyi Differential Privacy (Mironov 2017) and its relationship to (ϵ, δ)-DP.

Next lemma shows the post-processing property of (ϵ, δ)-DP, i.e., the composition of a data independent mapping f with an (ϵ, δ)-DP mechanism $\mathcal{M}$ also satisfies (ϵ, δ)-DP.

Lemma 2.7 ((Dwork and Roth 2014))**.** Consider a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ that is (ϵ, δ)-DP. Let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary randomized mapping. Then $f(\mathcal{M}) : \mathcal{S}^n \to \mathcal{R}'$ is (ϵ, δ)-DP.

3 The Proposed Algorithm

In this section, we present our differentially private sparse learning framework, which is illustrated in Algorithm 1. Note that Algorithm 1 will call IGHT algorithm (Yuan, Li, and Zhang 2014; Jain, Tewari, and Kar 2014) in Algorithm 2. IGHT enjoys linear convergence rate and is widely used for sparse learning. Note that for the sparsity constraint, i.e., $\|\boldsymbol{\theta}\|_0 \leq s$, the hard thresholding operator $\mathcal{H}_s(\boldsymbol{\theta})$ is defined as follows: $[\mathcal{H}_s(\boldsymbol{\theta})]_i = \theta_i$ if $i \in \mathrm{supp}(\boldsymbol{\theta}, s)$ and $[\mathcal{H}_s(\boldsymbol{\theta})]_i = 0$ otherwise, for $i \in [d]$. It preserves the largest s entries of $\boldsymbol{\theta}$ in magnitude. Equipped with IGHT, our framework also has a linear convergence rate for solving high-dimensional sparsity constrained problems.

Algorithm 1 Differentially Private Sparse Learning via Knowledge Transfer (DPSL-KT)

Input Loss function $\bar{L}_S$, distribution $\widetilde{\mathcal{D}}$, IGHT parameters $s, \eta_1, \eta_2, T_1, T_2$, function f, $\boldsymbol{\theta}_0$, σ

1: $\widehat{\boldsymbol{\theta}} = \mathrm{IGHT}(\boldsymbol{\theta}_0, \bar{L}_S, s, \eta_1, T_1)$
2: Generate training set: $S^{\mathrm{p}} = \{(\widetilde{\mathbf{x}}_i, y_i^{\mathrm{p}})\}_{i=1}^m$, where $y_i^{\mathrm{p}} = \langle \widehat{\boldsymbol{\theta}}, \widetilde{\mathbf{x}}_i \rangle + \xi_i$, $\widetilde{\mathbf{x}}_i \sim \widetilde{\mathcal{D}}$, $\xi_i \sim N(0, \sigma^2)$
3: Constructing the new task: $\widetilde{L}(\boldsymbol{\theta}) = (2m)^{-1} \sum_{i=1}^m \big(y_i^{\mathrm{p}} - \langle \boldsymbol{\theta}, \widetilde{\mathbf{x}}_i \rangle\big)^2$
4: $\boldsymbol{\theta}^{\mathrm{p}} = \mathrm{IGHT}(\boldsymbol{\theta}_0, \widetilde{L}, s, \eta_2, T_2)$

Output $\boldsymbol{\theta}^{\mathrm{p}}$

Algorithm 2 Iterative Gradient Hard Thresholding (IGHT)

Input Loss function L_S, parameters s, η, T, $\boldsymbol{\theta}_0$

1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: $\quad \boldsymbol{\theta}_t = \mathcal{H}_s\big(\boldsymbol{\theta}_{t-1} - \eta \nabla L_S(\boldsymbol{\theta}_{t-1})\big)$
3: **end for**

Output $\boldsymbol{\theta}_T$

There are two key ingredients in our framework: (1) an efficient problem solver, i.e., iterative gradient hard thresholding (IGHT) algorithm (Yuan, Li, and Zhang 2014; Jain, Tewari, and Kar 2014), and (2) the knowledge transfer procedure. In detail, we first solve the optimization problem (1.1) using IGHT, which is demonstrated in Algorithm 2, to get a non-private "teacher" estimator $\widehat{\boldsymbol{\theta}}$. The next step is the knowledge transfer procedure: we draw some synthetic features $\{\widetilde{\mathbf{x}}_i\}_{i=1}^m$ from a given distribution $\widetilde{\mathcal{D}}$, and output the corresponding private-preserving responses $\{y_i^{\mathrm{p}}\}_{i=1}^m$ using the Gaussian mechanism: $y_i^{\mathrm{p}} = \langle \widehat{\boldsymbol{\theta}}, \widetilde{\mathbf{x}}_i \rangle + \xi_i$, where ξ_i is the Gaussian noise to protect the private information contained in $\widehat{\boldsymbol{\theta}}$. Finally, by solving a new sparsity constrained learning problem $\widetilde{L}$ using the privacy-preserving synthetic dataset $S^{\mathrm{p}} = \{(\widetilde{\mathbf{x}}_i, y_i^{\mathrm{p}})\}_{i=1}^m$, we can get a differentially private "student" estimator $\boldsymbol{\theta}^{\mathrm{p}}$.

Our proposed knowledge transfer framework can achieve both strong privacy and utility guarantees. Intuitively speaking, the newly constructed learning problem can reduce the utilization of the privacy budget since we only require the generated responses to preserve the privacy of original training sample, which in turn leads to a strong privacy guarantee. In addition, this new learning problem contains the knowledge of the "teacher" estimator, which preserves the sparsity information of the underlying parameter. As a result, the "student" estimator can also have a strong utility guarantee.

4 Main Results

In this section, we will present the privacy and utility guarantees for Algorithm 1. We start with two conditions, which will be used in the result for generic models. Later, when we apply our result to specific models, these conditions will be verified explicitly.

The first condition is about the upper bound on the gradient of the function L_S, which will be used to characterize the statistical error of generic sparse models.

Condition 4.1. For a given sample size n and tolerance parameter $\zeta \in (0, 1)$, let $\varepsilon(n, \zeta)$ be the smallest scalar such that with probability at least $1 - \zeta$, we have $\|\nabla L_S(\boldsymbol{\theta}^*)\|_\infty \leq \varepsilon(n, \zeta)$.

To derive the utility guarantee, we also need the sparse eigenvalue condition (Zhang 2010) on the function L_S, which directly implies the restricted strong convex and smooth properties (Negahban et al. 2009; Loh and Wainwright 2013) of the function L_S.

Condition 4.2. The empirical loss L_S on the training data satisfies the sparse eigenvalue condition, if for all $\boldsymbol{\theta}$, there exist positive numbers μ and β such that

$$\mu = \inf_{\mathbf{v}} \big\{\mathbf{v}^\top \nabla^2 L_S(\boldsymbol{\theta}) \mathbf{v} \mid \|\mathbf{v}\|_0 \leq s,\ \|\mathbf{v}\|_2 = 1\big\},$$
$$\beta = \sup_{\mathbf{v}} \big\{\mathbf{v}^\top \nabla^2 L_S(\boldsymbol{\theta}) \mathbf{v} \mid \|\mathbf{v}\|_0 \leq s,\ \|\mathbf{v}\|_2 = 1\big\}.$$

4.1 Results for Generic Models

We first present the privacy guarantee of Algorithm 1 in the setting of (ϵ, δ)-DP.

Theorem 4.3. Suppose the loss function on each training example satisfies $\|\nabla \ell_i(\boldsymbol{\theta}_{\min})\|_\infty \leq \gamma$, and $\widetilde{\mathcal{D}}$ is a sub-Gaussian distribution with parameter $\widetilde{\alpha}$ and the covariance matrix $\|\widetilde{\boldsymbol{\Sigma}}\|_2 \leq \widetilde{\beta}$, and $m \geq C_1 \widetilde{\alpha} s \log d$ for some absolute constant C_1. Given a privacy budget ϵ and a constant $\delta \in (0, 1)$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies (ϵ, δ)-DP if $\sigma^2 = 8m\widetilde{\beta} s \gamma^2 \log(2.5/\delta)/(n^2 \epsilon^2 \lambda^2)$.

Remark 4.4. Theorem 4.3 suggests that in order to ensure the privacy guarantee, the only condition on the private data is the ℓ_∞-norm bound on the gradient of the loss function on each training example. This is in contrast to the ℓ_2-norm bound required by many previous work (Kifer, Smith, and Thakurta 2012; Talwar, Thakurta, and Zhang 2015; Wang and Gu 2019) for sparse learning problems. We remark that ℓ_∞-norm bound is a milder condition than ℓ_2-norm bound, and gives a better utility guarantee that only depends on the ℓ_∞-norm of the input data vectors instead of their ℓ_2-norm.

Next, we provide the linear convergence rate and the utility guarantee of Algorithm 1.

Theorem 4.5. Suppose that the loss function $\bar{L}_S$ is $\bar{\beta}$-smooth and L_S satisfies Condition 4.1 with parameter $\varepsilon(n,\zeta)$. Under the same conditions of Theorem 4.3 on ℓ_i, $\widetilde{\mathcal{D}}$, σ^2, there exist constants $\{C_i\}_{i=1}^8$ such that if $n = m \geq C_1\widetilde{\alpha}s\log d$, $s \geq C_2\kappa^2 s^*$ with $\kappa = \bar{\beta}/\lambda$, the stepsize $\eta_1 = C_3\lambda/\bar{\beta}^2, \eta_2 = C_4/\widetilde{\beta}$, then $\boldsymbol{\theta}^{\mathrm{p}}$ converges to $\boldsymbol{\theta}^*$ at a linear rate. In addition, if we choose $\lambda^2 = C_5\gamma\sqrt{s^*\log d\log(1/\delta)}/(n\epsilon)$, for large enough T_1, T_2, with probability at least $1-\zeta-C_6/d$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies

$$\|\boldsymbol{\theta}^{\mathrm{p}} - \boldsymbol{\theta}^*\|_2^2 \leq C_7\frac{s^*}{\bar{\beta}^2}\varepsilon(n,\zeta)^2 + C_8\big(1/\bar{\beta}^2 + \widetilde{\alpha}^2/\widetilde{\beta}\big)\frac{\gamma\sqrt{s^{*3}\log d\log(1/\delta)}}{n\epsilon}.$$

Remark 4.6. The utility bound of our method consists of two terms: the first term denotes the statistical error of generic sparse models, while the second one corresponds to the error introduced by the Gaussian mechanism, and is the dominating term. Therefore, the utility bound is of order $O\big(\gamma\sqrt{s^{*3}\log d\log(1/\delta)}/(n\epsilon)\big)$, which depends on the true sparsity s^* rather than the dimension of the problem d, and therefore is desirable for sparse learning.

The following corollary shows that if L_S further satisfies Condition 4.2, our method can achieve an improved utility guarantee.

Corollary 4.7. Suppose that L_S satisfies Condition 4.2 with parameters μ, β. Under the same conditions of Theorem 4.5 on $L_S, \ell_i, \widetilde{\mathcal{D}}$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies (ϵ,δ)-DP if we set $\lambda = 0$ and $\sigma^2 = 8m\widetilde{\beta}s\gamma^2\log(2.5/\delta)/(n^2\epsilon^2\mu^2)$. In addition, there exist constants $\{C_i\}_{i=1}^7$ such that if $n = m \geq C_1\widetilde{\alpha}s\log d$, $s \geq C_2\kappa^2 s^*$ with $\kappa = \beta/\mu$, step size $\eta_1 = C_3\mu/\beta^2, \eta_2 = C_4/\widetilde{\beta}$, for large enough T_1, T_2, with probability at least $1-\zeta-C_5/d$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies

$$\|\boldsymbol{\theta}^{\mathrm{p}} - \boldsymbol{\theta}^*\|_2^2 \leq C_6\frac{s^*}{\beta^2}\varepsilon(n,\zeta)^2 + C_7\widetilde{\alpha}^2\frac{\gamma^2 s^{*2}\log d\log(1/\delta)}{\widetilde{\beta}\mu^2 n^2\epsilon^2}.$$

Remark 4.8. Corollary 4.7 shows that if the training loss on the private data satisfies the sparse eigenvalue condition, Algorithm 1 can achieve $\widetilde{O}\big(\gamma^2 s^{*2}/(n\epsilon)^2\big)$ utility guarantee by setting $\lambda = 0$ and the variance σ^2 accordingly. It improves the utility without the sparse eigenvalue condition $\widetilde{O}\big(\gamma s^{*3/2}/(n\epsilon)\big)$ in Theorem 4.5 by a factor of $\widetilde{O}\big(n\epsilon/\gamma\sqrt{s^*}\big)$. Note that sparse eigenvalue condition has been verified for many sparse models (Negahban et al. 2009) including sparse linear regression and sparse logistic regression.

4.2 Results for Specific Models

In this subsection, we demonstrate the results of our framework for specific models. Note that the privacy guarantee has been established in Theorem 4.3, and we only present the utility guarantees.

Sparse linear regression We consider the following linear regression problem in the high-dimensional regime (Tibshirani 1996): $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}^* + \boldsymbol{\xi}$, where $\mathbf{y} \in \mathbb{R}^n$ is the response vector, $\mathbf{X} \in \mathbb{R}^{n\times d}$ denotes the design matrix, $\boldsymbol{\xi} \in \mathbb{R}^n$ is a noise vector, and $\boldsymbol{\theta}^* \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}^*\|_0 \leq s^*$ is the underlying sparse coefficient vector that we want to recover. In order to estimate the sparse vector $\boldsymbol{\theta}^*$, we consider the following sparsity constrained estimation problem, which has been studied in many previous work (Zhang 2011; Foucart and Rauhut 2013; Yuan, Li, and Zhang 2014; Jain, Tewari, and Kar 2014; Chen and Gu 2016)

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d}(2n)^{-1}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2/2 \ \text{ subject to } \ \|\boldsymbol{\theta}\|_0 \leq s. \tag{4.1}$$

The utility guarantee of Algorithm 1 for solving (4.1) can be implied by Theorem 4.5. Here we only need to verify Condition 4.2 for the sparse linear regression model. In specific, we can show that $\nabla L_S(\boldsymbol{\theta}^*) = \mathbf{X}^\top\boldsymbol{\xi}/n$, and we can prove that $\|\nabla L_S(\boldsymbol{\theta}^*)\|_\infty \leq C_1\nu\sqrt{\log d/n}$ holds with probability at least $1-\exp(-C_2 n)$, where C_1, C_2 are absolute constants. Therefore, we have $\zeta = 1-\exp(-C_2 n)$, $\varepsilon(n,\zeta) = C_1\nu\sqrt{\log d/n}$. By substituting these quantities into Theorem 4.5, we can obtain the following corollary.

Corollary 4.9. Suppose that each row of the design matrix satisfies $\max_{i\in[n]}\|\mathbf{x}_i\|_\infty \leq K$, and the noise vector $\boldsymbol{\xi} \sim N(0,\nu^2\mathbf{I}_n)$. Under the same conditions of Theorem 4.5 on $\widetilde{\mathcal{D}}, \sigma^2, \eta_1, \eta_2, s$, there exist constants $\{C_i\}_{i=1}^5$ such that if $m = n \geq C_1 s\log d$, $\lambda^2 = C_2K^2s^*\sqrt{\log d\log(1/\delta)}/(n\epsilon)$, with probability at least $1-C_3/d$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies

$$\|\boldsymbol{\theta}^{\mathrm{p}} - \boldsymbol{\theta}^*\|_2^2 \leq C_4\nu^2K^2\frac{s^*\log d}{n} + C_5\widetilde{\alpha}^2K^2\frac{s^{*2}\sqrt{\log d\log\frac{1}{\delta}}}{\widetilde{\beta}n\epsilon}.$$

Remark 4.10. Corollary 4.9 suggests that $O\big(s^*\log d/n + K^2s^{*2}\sqrt{\log d\log(1/\delta)}/(n\epsilon)\big)$ utility guarantee can be achieved by our algorithm. The term $O(s^*\log d/n)$ denotes the statistical error for sparse vector estimation, which matches the minimax lower bound (Raskutti, Wainwright, and Yu 2011). While the term $\widetilde{O}(K^2s^{*2}/(n\epsilon))$ corresponds to the error introduced by the privacy-preserving mechanism, and is the dominating term. Compared with the best-known result (Kifer, Smith, and Thakurta 2012; Wang and Gu 2019) $\widetilde{O}(\widetilde{K}^2s^{*2}/(n^2\epsilon^2))$, where $\|\mathbf{x}_i\|_2 \leq \widetilde{K}$ for all $i \in [n]$, our utility guarantee does not require the sparse eigenvalue condition and is better than their results by a factor of $\widetilde{O}\big(\widetilde{K}^2/(K^2n\epsilon)\big)$. Since we have $\widetilde{K} \leq \sqrt{d}K$ in the worst case, the improvement factor can be as large as $\widetilde{O}\big(d/(n\epsilon)\big)$. Compared with the utility guarantee $\widetilde{O}\big(1/(n\epsilon)^{2/3}\big)$ obtained by (Talwar, Thakurta, and Zhang 2015), our method improves their result by a factor of $\widetilde{O}\big((n\epsilon)^{1/3}/(Ks^*)^2\big)$, which demonstrates the advantage of our framework.

Remark 4.11. According to Corollary 4.7, if L_S satisfies Condition 4.2 with parameters μ, β, we can set $\lambda = 0$ and

$\sigma^2 = 8s\gamma^2\log(2.5/\delta)/(n\epsilon^2\mu^2)$ in Algorithm 1. As a result, the output of Algorithm 1 will satisfy (ϵ, δ)-DP with the utility guarantee $\widetilde{O}\big(K^2 s^{*3}/(n^2\epsilon^2)\big)$, which improves the result in Corollary 4.9 by a factor of $\widetilde{O}\big(n\epsilon/s^*\big)$. Due to the space limit, we defer the detailed result to the supplemental material.

Sparse logistic regression For high-dimensional logistic regression, we assume the label of each example follows an i.i.d. Bernoulli distribution conditioned on the input vector $\mathbb{P}(y = 1|\mathbf{x}, \boldsymbol{\theta}^*) = \exp\big(\boldsymbol{\theta}^{*\top}\mathbf{x} - \log\big(1 + \exp(\boldsymbol{\theta}^{*\top}\mathbf{x})\big)\big)$, where $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $\boldsymbol{\theta}^* \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}^*\|_0 \leq s^*$ is the sparse parameter vector we would like to estimate. Given observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we consider the following maximum likelihood estimation problem with sparsity constraints (Yuan, Li, and Zhang 2014; Chen and Gu 2016)

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} -n^{-1}\textstyle\sum_{i=1}^n \big[y_i\boldsymbol{\theta}^\top\mathbf{x}_i - \log\big(1 + \exp(\boldsymbol{\theta}^\top\mathbf{x}_i)\big)\big] \quad (4.2)$$

$$+ \lambda\|\boldsymbol{\theta}\|_2^2/2 \text{ subject to } \|\boldsymbol{\theta}\|_0 \leq s. \quad (4.3)$$

The utility guarantee of Algorithm 1 for solving (4.2) is shown in the following corollary.

Corollary 4.12. Under the same conditions of Corollary 4.9 on $\mathbf{x}_i, \widetilde{\mathcal{D}}, \sigma^2, \eta_1, \eta_2, s$, there exist constants $\{C_i\}_{i=1}^5$ such that if $m = n \geq C_1 s\log d$, $\lambda^2 = C_2 K\sqrt{s^*\log d\log(1/\delta)/(n\epsilon)}$, with probability at least $1 - C_3/d$, the output $\boldsymbol{\theta}^{\mathrm{p}}$ of Algorithm 1 satisfies

$$\|\boldsymbol{\theta}^{\mathrm{p}} - \boldsymbol{\theta}^*\|_2^2 \leq C_4 K^2\frac{s^*\log d}{n} + C_5\widetilde{\alpha}^2 K\frac{\sqrt{s^{*3}\log d\log(1/\delta)}}{\widetilde{\beta}n\epsilon}.$$

Remark 4.13. Corollary 4.12 suggests that $O\big(s^*\log d/n + K\sqrt{s^{*3}\log d\log(1/\delta)/(n\epsilon)}\big)$ utility guarantee can be obtained by our algorithm for sparse logistic regression. The term $\widetilde{O}\big(Ks^{*3/2}/(n\epsilon)\big)$ caused by the Gaussian mechanism is the dominating term and does not depend on the sparse eigenvalue condition, and is better than the best-known result (Wang and Gu 2019) $\widetilde{O}\big(\widetilde{K}^2 s^{*2}/(n^2\epsilon^2)\big)$ by a factor of $\widetilde{O}\big(\widetilde{K}^2 s^{*1/2}/(Kn\epsilon)\big)$. The improvement factor can be as large as $\widetilde{O}\big(dK/(n\epsilon)\big)$ since $\widetilde{K} \leq \sqrt{d}K$.

5 Numerical Experiments

In this section, we present experimental results of our proposed algorithm on both synthetic and real datasets. For sparse linear regression, we compare our framework with Two stage (Kifer, Smith, and Thakurta 2012), Frank-Wolfe (Talwar, Thakurta, and Zhang 2015), and DP-IGHT (Wang and Gu 2019) algorithms. For sparse logistic regression, we compare our framework with DP-IGHT (Wang and Gu 2019) algorithm. For all of our experiments, we choose the parameters of different methods according to the requirements of their theoretical guarantees. More specifically, on the synthetic data experiments, we assume s^* is known for all the methods. On the real data experiments, s^* is unknown, neither our method or the competing methods has the knowledge of s^*. So we simply choose a sufficiently large s as a surrogate of s^*. Given s, for the parameter λ in our method, according to Theorem 4.5, we choose λ from a sequence of values $c_1\sqrt{s\log d\log(1/\delta)/(n\epsilon)}$, where $c_1 \in \{10^{-6}, 10^{-5}, \ldots, 10^1\}$, by cross-validation. For competing methods, given s, we choose the iteration number of Frank-Wolfe from a sequence of values $c_2 s$, where $c_2 \in \{0.5, 0.6, \ldots, 1.5\}$, and the regularization parameter in the objective function of Two Stage from a sequence of values $c_3 s/\epsilon$, where $c_3 \in \{10^{-3}, 10^{-2}, \ldots, 10^2\}$, by cross-validation. For DP-IGHT, we choose its stepsize from the grid $\{1/2^0, 1/2^1, \ldots, 1/2^6\}$ by cross-validation. For the non-private baseline, we use the non-private IGHT (Yuan, Li, and Zhang 2014).

5.1 Numerical Simulations

In this subsection, we investigate our framework on synthetic datasets for sparse linear and logistic regression. In both problems, we generate the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ such that each entry is drawn i.i.d. from a uniform distribution $U(-1, 1)$, and the underlying sparse vector $\boldsymbol{\theta}^*$ has s nonzero entries that are randomly generated. In addition, we consider the following two settings: (i) $n = 800, d = 1000, s^* = 10$; (ii) $n = 4000, d = 5000, s^* = 50$. We choose $\widetilde{\mathcal{D}}$ to be a uniform distribution $U(-1, 1)$, which implies $\widetilde{\beta} = 1/3$.

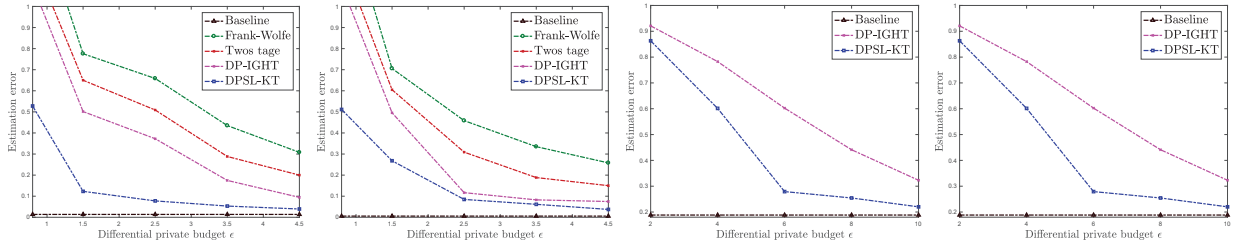
Sparse linear regression For sparse linear regression, the observations are generated according to the linear regression model $\mathbf{y} = \mathbf{X}^\top\boldsymbol{\theta}^* + \boldsymbol{\xi}$, where the noise vector $\boldsymbol{\xi} \sim N(0, \nu^2\mathbf{I})$ with $\nu^2 = 0.1$. In our experiments, we set $\delta = 0.01$ and vary the privacy budget ϵ from 0.8 to 5. Note that due to the hardness of the problem itself, we choose relatively large privacy budgets compared with the low-dimensional problem to ensure meaningful results. Figure 2(a) and 2(b) illustrate the estimation error $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2/\|\boldsymbol{\theta}^*\|_2$ of different methods averaged over 10 trails. The results show that the estimation error of our method is close to the non-private baseline, and is significantly better than other private baselines. Even when we have a small privacy budget (i.e., $\epsilon = 0.8$), our method can still recover the underlying sparse vector with reasonably small estimation error, while others fail.

Sparse logistic regression For sparse logistic regression, each label is generated from the logistic distribution $\mathbb{P}(y = 1) = 1/\big(1 + \exp(\mathbf{x}_i^\top\boldsymbol{\theta}^*)\big)$. In this problem, we vary the privacy budget ϵ from 2 to 10, and set $\delta = 0.01$. We present the estimation error versus privacy budget ϵ of different methods in Figure 2(c) and 2(d). The results show that our method can output accurate estimators when we have relative large privacy budget, and it consistently outperforms the private baseline.

5.2 Real Data Experiments

For real data experiments, we use E2006-TFIDF dataset (Kogan et al. 2009) and RCV1 dataset (Lewis et al. 2004), for the evaluation of sparse linear regression and sparse logistic regression, respectively.

E2006-TFIDF data For sparse linear regression problem, we use E2006-TFIDF dataset, which consists of financial



(a) Linear regression (b) Linear regression (c) Logistic regression (d) Logistic regression

Figure 2: Numerical results for sparse linear and logistic regression. (a), (b) Reconstruction error versus privacy budget for sparse linear regression; (c), (d) Reconstruction error versus privacy budget for sparse linear regression.

Table 2: Comparison of different algorithms for various privacy budgets ϵ with $\delta = 10^{-5}$ in terms of MSE (mean $\pm$ std) and its corresponding standard deviation on E2006-TFIDF.

Method	$\epsilon = 0.8$	$\epsilon = 1.5$	$\epsilon = 2.5$	$\epsilon = 3.5$	$\epsilon = 4.5$
IGHT	0.8541	0.8541	0.8541	0.8541	0.8541
Frank-Wolfe	4.471 (0.239)	2.004 (0.155)	1.535 (0.140)	1.206 (0.095)	1.099 (0.082)
Two stage	4.022 (0.159)	1.803 (0.141)	1.326 (0.093)	1.107 (0.103)	1.053 (0.069)
DP-IGHT	3.731 (0.207)	1.687 (0.126)	1.304 (0.035)	1.067 (0.051)	0.968 (0.062)
DPSL-KT	1.227 (0.110)	1.178 (0.056)	1.065 (0.054)	0.971 (0.031)	0.952 (0.010)

Table 3: Comparison of different algorithms for various privacy budgets ϵ with $\delta = 10^{-5}$ in terms of test error (mean $\pm$ std) and its corresponding standard deviation on RCV1 data.

Method	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$
IGHT	0.0645	0.0645	0.0645	0.0645
Frank-Wolfe	0.1381 (0.0045)	0.1134 (0.0041)	0.0978 (0.0032)	0.0882 (0.0033)
Two stage	0.1272 (0.0044)	0.1061(0.0038)	0.0949 (0.0035)	0.0866 (0.0031)
DP-IGHT	0.1179 (0.0035)	0.1026 (0.0036)	0.0922 (0.0032)	0.0824 (0.0029)
DPSL-KT	0.1105 (0.0038)	0.0974 (0.0035)	0.0885 (0.0029)	0.0787(0.0031)

risk data from thousands of U.S. companies. In detail, it contains 16087 training examples, 3308 testing examples, and we randomly sample 25000 features for this experiment. In order to validate our proposed framework, we randomly divide the original dataset into two datasets: private dataset and public dataset. For the private dataset, it contains 8044 training examples, and we assume that this dataset contains the sensitive information that we want to protect. For the public dataset, it contains 8043 training examples. We set $s = 2000$, $\delta = 10^{-5}$, $\epsilon \in [0.8, 5]$. We estimate $\widetilde{\beta}$ by the sample covariance matrix, and the detailed estimation procedure can be found in the longer version of this paper. Table 2 reports the mean square error (MSE) on the test data of different methods for various privacy budgets over 10 trails. The results show that the performance of our algorithm is close to the non-private baseline even when we have small private budgets, and is much better than existing methods.

RCV1 data For sparse logistic regression, we use a Reuters Corpus Volume I (RCV1) data set for text categorization research. RCV1 is released by Reuters, Ltd. for research purposes, and consists of over 800000 manually categorized newswire stories. It contains 20242 training examples, 677399 testing examples and 47236 features. As before, we randomly divide the original dataset into two datasets with equal size serving as the private and publice datasets. In addition, we randomly choose 10000 test examples and 20000 features, and set $s = 500$, $\delta = 10^{-5}$, $\epsilon \in [2, 8]$. We estimate $\widetilde{\beta}$ using the same method as before. We compare all algorithms in terms of their classification error on the test set over 10 replications, which is summarized in Table 3. Evidently our algorithm achieves the lowest test error among all private algorithms on RCV1 dataset, which demonstrates the superiority of our algorithm.

6 Conclusions and Future Work

In this paper, we developed a differentially private framework for sparse learning using the idea of knowledge transfer. We establish the linear convergence rate and the utility guarantee of our method. Experiments on both synthetic and real-world data demonstrate the superiority of our algorithm. For the future work, it is very interesting to generalize our framework to other structural constrained learning problems such as the low-rank estimation problem. It is also very interesting to study the theoretical lower-bound of the differentially private sparse learning problem to access the

optimality of our proposed method.

Acknowledgements

This research was sponsored in part by the National Science Foundation SaTC-1717950. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Bassily, R.; Smith, A.; and Thakurta, A. 2014. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. *arXiv preprint arXiv:1405.7085*.

Bassily, R.; Thakkar, O.; and Thakurta, A. 2018. Model-agnostic private learning via stability. *arXiv preprint arXiv:1803.05101*.

Bickel, P. J.; Ritov, Y.; Tsybakov, A. B.; et al. 2009. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics* 37(4):1705–1732.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535–541. ACM.

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

Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12(Mar):1069–1109.

Chen, J., and Gu, Q. 2016. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *UAI*.

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

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 265–284. Springer.

Foucart, S., and Rauhut, H. 2013. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel.

Frank, M., and Wolfe, P. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly* 3(1-2):95–110.

Hamm, J.; Cao, Y.; and Belkin, M. 2016. Learning privately from multiparty data. In *International Conference on Machine Learning*, 555–563.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jain, P., and Thakurta, A. G. 2014. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, 476–484.

Jain, P.; Tewari, A.; and Kar, P. 2014. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, 685–693.

Jayaraman, B.; Wang, L.; Evans, D.; and Gu, Q. 2018. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *NeurIPS*, 6346–6357.

Kifer, D.; Smith, A.; and Thakurta, A. 2012. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, 25–1.

Kogan, S.; Levin, D.; Routledge, B. R.; Sagi, J. S.; and Smith, N. A. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 272–280. Association for Computational Linguistics.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research* 5(Apr):361–397.

Loh, P.-L., and Wainwright, M. J. 2013. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*.

Mironov, I. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE.

Negahban, S.; Yu, B.; Wainwright, M. J.; and Ravikumar, P. K. 2009. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, 1348–1356.

Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

Raskutti, G.; Wainwright, M. J.; and Yu, B. 2011. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory* 57(10):6976–6994.

Talwar, K.; Thakurta, A. G.; and Zhang, L. 2015. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, 3025–3033.

Thakurta, A. G., and Smith, A. 2013. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, 819–850.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Vershynin, R. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Wang, L., and Gu, Q. 2019. Differentially private iterative gradient hard thresholding for sparse learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.

Wang, L.; Jayaraman, B.; Evans, D.; and Gu, Q. 2019. Efficient privacy-preserving nonconvex optimization. *arXiv preprint arXiv:1910.13659*.

Wang, D.; Gaboardi, M.; and Xu, J. 2018. Empirical risk minimization in non-interactive local differential privacy revisited. In *Advances in Neural Information Processing Systems*, 973–982.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, 2719–2728.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2018. Pate-gan: Generating synthetic data with differential privacy guarantees.

Yuan, X.; Li, P.; and Zhang, T. 2014. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning*, 127–135.

Zhang, J.; Zheng, K.; Mou, W.; and Wang, L. 2017. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*.

Zhang, T. 2010. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research* 11(Mar):1081–1107.

Zhang, T. 2011. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE transactions on information theory* 57(7):4689–4708.