Differentially Private Iterative Gradient Hard Thresholding for Sparse Learning

Lingxiao Wang , **Quanquan Gu**

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

Abstract

We consider the differentially private sparse learning problem, where the goal is to estimate the underlying sparse parameter vector of a statistical model in the high-dimensional regime while preserving the privacy of each training example. We propose a generic differentially private iterative gradient hard threshoding algorithm with a linear convergence rate and strong utility guarantee. We demonstrate the superiority of our algorithm through two specific applications: sparse linear regression and sparse logistic regression. Specifically, for sparse linear regression, our algorithm can achieve the best known utility guarantee without any extra support selection procedure used in previous work [Kifer *et al.*, 2012]. For sparse logistic regression, our algorithm can obtain the utility guarantee with a logarithmic dependence on the problem dimension. Experiments on both synthetic data and real world datasets verify the effectiveness of our proposed algorithm.

1 Introduction

In modern high-dimensional data analytics, where the problem dimension can increase with the number of observations, sparse learning has emerged as a prominent method to alleviate overfitting and provide statistically reliable results. Consequently, many sparse learning algorithms such as ℓ_1 convex relaxation based methods [Tibshirani, 1996; Van de Geer and others, 2008; Negahban *et al.*, 2009] have been proposed in the past two decades. Compared with ℓ_1 convex relaxation based sparse learning algorithms, ℓ_0 constrained sparse learning algorithms [Zhang, 2011; Yuan *et al.*, 2014; Jain *et al.*, 2014; Chen and Gu, 2016] received increasing attention due to its small estimation bias. In specific, the ℓ_0 constrained sparse learning is formulated as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} L_S(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n}\ell(\boldsymbol{\theta};\mathbf{z}_i) \text{ subject to } \|\boldsymbol{\theta}\|_0 \le s, \quad (1.1)$$

where $S = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$ denotes the training dataset with $\mathbf{z}_i = (\mathbf{x}_i, y_i)$, L_S is the empirical loss function, $\|\boldsymbol{\theta}\|_0$ denotes the number of nonzero entries in $\boldsymbol{\theta}$, s is a parameter for tuning the sparsity level of $\boldsymbol{\theta}$, and we assume that the data are generated from some underlying statistical model with sparse parameter vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$ such that $\|\boldsymbol{\theta}^*\|_0 = s^*$. The goal of sparse learning is to recover $\boldsymbol{\theta}^*$.

In many applications, the data used for sparse learning are sensitive datasets, such as financial records or genomic data, raising a big concern that the adversaries may be able to infer the private information from the trained model. This privacy concern necessitates the private-preserving algorithms for learning sparse models. The prerequisite for developing such algorithms is a rigorous privacy definition. In recent years, differential privacy [Dwork *et al.*, 2006] has been served as the most widely adopted notion of statistical data privacy and has been applied to many real world applications [Erlingsson *et al.*, 2014; Ding *et al.*, 2017]. The formal definition of differential privacy is as follows.

Definition 1.1 (Differential privacy [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 data sets $S, S' \in \mathcal{S}^n$ differing by one example, and any output subset $O \subseteq \mathcal{R}$, it holds that

$$\mathbb{P}[\mathcal{M}(S) \in O] \le e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(S') \in O] + \delta.$$

According to the definition, differential privacy requires that datasets differing by one example lead to similar distributions on the output of a randomized algorithm. This implies that an adversary will draw essentially the same conclusions about an individual whether or not that individual's data was used even if many records are known a priori to the adversary.

There exist several studies [Kifer *et al.*, 2012; Thakurta and Smith, 2013; Talwar *et al.*, 2015] trying to develop differentially private algorihthms for solving sparse learning problems. However, they only consider sparse linear regression, and the convergence rates and utility guarantees of these methods are suboptimal. In order to overcome the limitations of existing differentially private sparse learning algorithms, we propose a differentially private iterative gradient hard thresholding (DP-IGHT) algorithm for solving the sparsity constrained learning problem (1.1), which is not only very efficient but also has comparable or even better utility guarantees than the state-of-the-art methods. We summarize the contributions of our work as follows

- Compared with existing work that is limited to sparse linear regression, our differentially private sparse learning algorithm is generic enough that it can be applied to a broad family of loss functions that satisfy the restricted strong convexity and smoothness conditions [Bickel *et al.*, 2009; Negahban *et al.*, 2009], and each component function is Lipschitz continuous. We demonstrate the superiority of our framework through two concrete examples: sparse linear regression and sparse logistic regression.

- We prove the linear convergence rate for our DP-IGHT algorithm, which outperforms the sub-linear convergence rate of Frank-Wolfe based method [Talwar *et al.*, 2015], and does not rely on any computationally intractable support selection algorithm as required by [Kifer *et al.*, 2012].
- We establish strong utility guarantee for our DP-IGHT algorithm. Specifically, it achieves the best known utility guarantee [Kifer *et al.*, 2012] for sparse linear regression while not requiring any extra support selection procedure. Our approach also provides the first utility guarantee 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 supp($\mathbf{x}$) be the index set of nonzero entries of $\mathbf{x}$, and 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 and $\mathcal{R}$ to denote the output space. Given two sequences $\{a_n\}$ and $\{b_n\}$, 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 denote the d by d identity matrix by $\mathbf{I}_d$. For simplicity, we use $\ell_i(\cdot)$ to denote $\ell(\cdot; \mathbf{z}_i)$ throughout the paper.

2 Related Work

To develop differentially private algorithms, the commonly used methods include output perturbation [Chaudhuri and Monteleoni, 2009], objective perturbation [Chaudhuri and Monteleoni, 2009], and gradient (iterative) perturbation [Bassily *et al.*, 2014]. More specifically, output perturbation adds random noise to the output of a non-private algorithm. Objective perturbation perturbs the objective function of learning algorithms by random noise before learning. And the idea of gradient perturbation is to introduce random noise into the intermediate steps of the learning algorithm. Although these approaches have been extensively studied for empirical risk minimization [Chaudhuri and Monteleoni, 2009; Chaudhuri *et al.*, 2011; Kifer *et al.*, 2012; Bassily *et al.*, 2014; Zhang *et al.*, 2017; Wang *et al.*, 2017; 2018; Jayaraman *et al.*, 2018] in classical setting, their applications to sparse learning in the high-dimensional regime remain understudied.

There exist several ad hoc approaches [Kifer *et al.*, 2012; Thakurta and Smith, 2013; Jain and Thakurta, 2014; Talwar *et al.*, 2015] to solving differentially private (sparse) learning in the high-dimensional setting. For example, [Jain and Thakurta, 2014] proposed a differentially private algorithm with the dimension independent utility guarantee for empirical risk minimization. Nevertheless, their method only works for specific loss functions and the utility guarantee is sub-optimal in terms of other parameters. The most relevant studies to ours are [Kifer *et al.*, 2012; Thakurta and Smith, 2013; Talwar *et al.*, 2015], which studied differentially private sparse linear regression. In detail, [Kifer *et al.*, 2012; Thakurta and Smith, 2013] proposed to first perform some differentially private model selection algorithms to estimate the support set of sparse model parameter vector, and then run the objective perturbation algorithm to estimate the parameter vector with its support restricted to the estimated subset. While, their method can achieve $O\big(s^{*2}\log(2/\gamma)/(n\epsilon)^2\big)$ utility guarantee, where ϵ is the privacy budget and γ is the probability that the model selection algorithms can successfully select the true support, the model selection algorithms, such as exponential mechanism, may be computational inefficient or even intractable in practice. In addition, the privacy and utility guarantees of their algorithm only holds for the exact optimal solution to the perturbed optimization problem. Later on, [Talwar *et al.*, 2015] developed a differentially private Frank-Wolfe algorithm, which is based on the gradient perturbation, for sparse learning. They showed that their algorithm can achieve $O\big(\omega(\mathcal{C})^{2/3}\log n/(n\epsilon)^{3/2}\big)$ utility guarantee, where $\omega(\mathcal{C})$ is the Gaussian width of the constraint set $\mathcal{C}$. However, their approach only has a sublinear convergence rate and the Gaussian width $\omega(\mathcal{C})$ can only be estimated for some specific convex set such as ℓ_1-norm ball.

Different from the aforementioned methods for sparse learning, our proposed DP-IGHT algorithm does not require exactly solving the optimization problem or the extra model selection procedure. Therefore, it is able to attain better empirical performances and more preferable in practice. In addition, our algorithm enjoys a linear convergence rate, which is more efficient than previous methods. The detailed comparisons of different algorithms for sparse linear regression are summarized in Table 1.

3 Preliminaries

In this section, we present some definitions that will be used throughout our paper. We first introduce several classes of functions that we considered in our work.

Definition 3.1 (G-Lipschitz continuous)**.** A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is G-Lipschitz continuous, if the following inequality holds for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \text{dom} f$

$$|f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2)| \leq G\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2.$$

Note that for a differentiable function f, G-Lipschitz continuous implies that the gradient norm is bounded, i.e., $\|\nabla f(\boldsymbol{\theta})\|_2 \leq G$ for all $\boldsymbol{\theta} \in \text{dom} f$.

Definition 3.2 (Sparse eigenvalue condition)**.** A twice differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies sparse eigenvalue condition with parameters $\mu > 0$ and $\beta > 0$, if the following holds for the Hessian of f for all $\boldsymbol{\theta} \in \text{dom} f$,

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

For sparse learning problems, sparse eigenvalue condition [Bickel *et al.*, 2009] implies the restricted strong convexity and smoothness conditions [Negahban *et al.*, 2009; Loh and

Algorithm	Method	Utility	Convergence	RSC/RSS	Support selection
Frank-Wolfe [Talwar *et al.*, 2015]	Iterative	$O\left(\frac{\log(nd)}{(n\epsilon)^{2/3}}\right)$	Sub-linear	No	No
Two stage [Kifer *et al.*, 2012]	Objective	$O\left(\frac{s^{*2}\log(2/\gamma)}{(n\epsilon)^2}\right)$	NA	Yes	Yes
DP-IGHT This paper	Iterative	$O\left(\frac{s^{*2}\log d}{(n\epsilon)^2}\right)$	Linear	Yes	No

Table 1: Comparison of different (ϵ, δ)-DP algorithms for sparse linear regression. We ignore the $\log(1/\delta)$ term in the utility guarantees. Note that γ is the probability that the differentially private model selection algorithms can successfully recover the true support.

Wainwright, 2013], which guarantee the objective function behaves like a strongly convex and smooth function over a sparse domain even the function is general convex in its entire domain. In the following discussion, we denote κ by β/μ.

Zero-concentrated differential privacy. Although the notion of (ϵ, δ)-DP, i.e., Definition 1.1, is widely used for the analysis of the output and objective perturbation methods, it is not suitable for the gradient perturbation method since it will give loose composition results. We propose to use the notion of zero-concentrated differential privacy [Bun and Steinke, 2016], which has a sharp composition result and thus is a better choice for gradient perturbation method.

Definition 3.3 (Zero-concentrated differential privacy)**.** A randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies ρ-zero-concentrated differential privacy (ρ-zCDP) if for any two adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one example, it holds that for all $\alpha \in (1, \infty)$

$$D_\alpha(\mathcal{M}(S)||\mathcal{M}(S')) \leq \rho\alpha, \tag{3.1}$$

where $D_\alpha(\mathcal{M}(S)||\mathcal{M}(S'))$ is the α-Renyi divergence[1] between two distributions $\mathcal{M}(S)$ and $\mathcal{M}(S')$.

Note that ρ-zCDP can be converted to (ϵ, δ)-DP through the following lemma, which is established in [Bun and Steinke, 2016].

Lemma 3.4. If a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies ρ-zCDP, then it satisfies $(\rho + 2\sqrt{\rho\log(1/\delta)}, \delta)$-differential privacy for any $\delta > 0$.

Next, we introduce the definition of ℓ_2-sensitivity, which is used to control the variance of the Gaussian Mechanism for ensuring ρ-zCDP.

Definition 3.5 (ℓ_2-sensitivity [Dwork *et al.*, 2006])**.** 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 $\Delta_2(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$.

Based on ℓ_2-sensitivity, we can use Gaussian mechanism to make our algorithms satisfy ρ-zCDP.

Lemma 3.6 (Gaussian mechanism [Bun and Steinke, 2016])**.** Given a function $q : \mathcal{S}^n \to \mathbb{R}^d$, the Gaussian Mechanism $\mathcal{M}(S) = q(S) + \mathbf{u}$, where $\mathbf{u} \sim N(0, \sigma^2\mathbf{I})$, satisfies $\Delta_2(q)^2/(2\sigma^2)$-zCDP.

ρ-zCDP has the invariant property of post-processing and the property of composition as follows.

Lemma 3.7 ([Bun and Steinke, 2016])**.** For two randomized mechanisms $\mathcal{M}_1 : \mathcal{S}^n \to \mathbb{R}^d$, $\mathcal{M}_2 : \mathcal{S}^n \times \mathbb{R}^d \to \mathbb{R}^d$. If $\mathcal{M}_1$ satisfies ρ_1-zCDP and $\mathcal{M}_2$ satisfies ρ_2-zCDP, then $\mathcal{M}_2(S, M_1(S))$ satisfies $(\rho_1 + \rho_2)$-zCDP.

4 Algorithmic Framework

In this section, we present our differentially private iterative gradient hard thresholding (DP-IGHT) algorithm, which is illustrated in Algorithm 1, for solving the sparsity constrained optimization problem (1.1).

Algorithm 1 Differentially Private Iterative Gradient Hard Thresholding (DP-IGHT)

```
Input: loss function $L_S$, thresholding parameters $s$, step size
  $\eta$, iteration number $T$, initial estimator $\boldsymbol{\theta}_0$, privacy budget
  $\rho$, Lipschitz constant $G$
  for $t = 1, 2, 3, \ldots, T$ do
    $\boldsymbol{\theta}_t = \mathcal{H}_s\big(\boldsymbol{\theta}_{t-1} - \eta(\nabla L_S(\boldsymbol{\theta}_{t-1}) + \mathbf{u})\big)$, where $\mathbf{u} \sim$
    $N(0, \sigma^2\mathbf{I}_d)$ with $\sigma^2 = TG^2/(n^2\rho)$
  end for
Output: $\boldsymbol{\theta}_T$
```

At the core of our proposed Algorithm 1 is the gradient perturbation procedure at each iteration, which ensures the differential privacy. More specifically, we perturb the gradient with Gaussian noise at each iteration and make use of the composition and post-processing properties of differential privacy to characterize the upper bound on the total privacy loss. Compared with the objective perturbation based approaches [Chaudhuri and Monteleoni, 2009; Chaudhuri *et al.*, 2011; Kifer *et al.*, 2012], our algorithm does not require the optimization problem to be solved exactly in order to achieve the privacy and utility guarantees. In addition, since it is very hard to characterize the sensitivity of the optimization problem with the sparsity constraint [Xu *et al.*, 2012], we do not pursue the output perturbation based approaches [Chaudhuri *et al.*, 2011; Zhang *et al.*, 2017].

According to Algorithm 1, to enforce the sparsity constraint, we use the iterative gradient hard thresholding (IGHT) algorithm, which has been shown to have a linear rate of convergence [Jain *et al.*, 2014; Yuan *et al.*, 2014; Chen and Gu, 2016]. Note that if we set $\sigma^2 = 0$ in Algorithm 1, it will reduce to the original IGHT algorithm. The hard thresholding operator $\mathcal{H}_s(\cdot)$ in Algorithm 1 is defined as follows:

$$[\mathcal{H}_s(\boldsymbol{\theta})]_i = \begin{cases} \theta_i, & \text{if } i \in \text{supp}(\boldsymbol{\theta}, s) \\ 0, & \text{otherwise .} \end{cases} \tag{4.1}$$

[1] The formal definition can be found in [Rényi, 1961].

$\mathcal{H}_s(\boldsymbol{\theta})$ preserves the top s elements in $\boldsymbol{\theta}$ in terms of magnitude and set others to be zero. We will show later that the proposed DP-IGHT algorithm also enjoys a linear convergence rate, and therefore is more efficient than existing methods.

5 Main Results

In this section, we first present the main theoretical properties of Algorithm 1 for generic models, and then show its applications to two specific examples: sparse linear regression and sparse logistic regression.

Recall that the goal of sparse learning is to estimate the underlying sparse parameter vector $\boldsymbol{\theta}^*$ of a statistical model. Thus, we impose a high probability upper bound on the gradient of the objective function at $\boldsymbol{\theta}^*$, which is used to characterize the statistical error of different statistical models.

Condition 5.1. For a given sample size n and tolerance parameter $\gamma \in (0, 1)$, let $\varepsilon(n, \gamma)$ be the smallest scalar such that with probability at least $1 - \gamma$, we have

$$\|\nabla L_S(\boldsymbol{\theta}^*)\|_\infty \le \varepsilon(n, \gamma),$$

where $\varepsilon(n, \gamma)$ depends on the sample size n and γ.

Equipped with this condition, we are ready to establish the main theoretical results of Algorithm 1.

5.1 Results for Generic Model

We first present the privacy guarantee of Algorithm 1 for solving sparse learning problem (1.1) under ρ-zCDP.

Theorem 5.2. Suppose each component function ℓ_i of L_S is G-Lipschitz continuous, the output $\boldsymbol{\theta}_T$ of Algorithm 1 satisfies ρ-zCDP after T iterations if $\sigma^2 = TG^2/(n^2\rho)$.

Remark 5.3. According to Lemma 3.4, we can also derive that the output $\boldsymbol{\theta}_T$ of Algorithm 1 satisfies (ϵ, δ)-DP if $\sigma^2 = TG^2/\big(n\big(\sqrt{\log(1/\delta) + \epsilon} - \sqrt{\log(1/\delta)}\big)\big)^2$. Furthermore, if $\epsilon \le \log(1/\delta)$, we can get $\sigma^2 \le 6TG^2\log(1/\delta)/(n\epsilon)^2$, which matches the bound of the noise variance for gradient perturbation methods [Wang *et al.*, 2017]. Note that for the Lipschitz parameter G, we can exactly calculate a tight upper bound for sparse linear regression and sparse logistic regression. However, for general loss functions, one practical approach to choose G is to use gradient clipping [Abadi *et al.*, 2016].

Next, we provide the utility guarantee of Algorithm 1 for solving sparse learning problem (1.1).

Theorem 5.4. Suppose the loss function L_S satisfies sparse eigenvalue condition with parameters μ, β, and Condition 5.1, and each component function ℓ_i is G-Lipschitz continuous. There exist constants $\{C_i\}_{i=1}^5$ such that if $\sigma^2 = TG^2/(n^2\rho)$, $\eta = C_1/(\beta + \mu)$, and $s \ge C_2\kappa^2 s^*$, then $\boldsymbol{\theta}_T$ converges to $\boldsymbol{\theta}^*$ at a linear rate. In addition, if we choose $T = C_3\kappa\log\big(\rho n^2\mu^2\|\boldsymbol{\theta}^*\|_2^2/(\kappa^2G^2 s\log d)\big)$, the following holds with probability at least $1 - \gamma$

$$\mathbb{E}\|\boldsymbol{\theta}_T - \boldsymbol{\theta}^*\|_2^2 \le C_4\frac{\kappa^2 s^*}{\mu^2}\varepsilon(n, \gamma)^2 + C_5\frac{\kappa^3G^2s^*\log d}{n^2\mu^2\rho}\cdot\log\frac{\rho n\mu\|\boldsymbol{\theta}^*\|_2}{s^*\kappa G}, \quad (5.1)$$

where the expectation is taken over the randomness of the Gaussian noises in Algorithm 1.

Remark 5.5. The utility bound in (5.1) consists of two terms: the first one denotes the statistical error, while the second term corresponds to the error introduced by the Gaussian mechanism. It is worth noting that the error term caused by the Gaussian mechanism depends on $s^*\log d$ instead of d comparing with the previous differentially private learning algorithms [Bassily *et al.*, 2014]. According to Lemma 3.4, we can also derive the following utility guarantee under (ϵ, δ)-DP

$$O\big(s^*\kappa^2\varepsilon(n, \gamma)^2/\mu^2 + \kappa^3G^2s^*\log d\log(1/\delta)/(n\epsilon\mu)^2\big),$$

and we defer such result to the supplemental material.

5.2 Implications for Specific Examples

In this subsection, we demonstrate the implications of the main theory for Algorithm 1 when it is applied to specific examples. Note that here we directly spell out the utility results under (ϵ, δ)-DP for the ease of comparison.

Sparse Linear Regression

The first example we considered is the linear regression problem in the high-dimensional regime $y_i = \langle\mathbf{x}_i, \boldsymbol{\theta}^*\rangle + \xi_i$, where $\mathbf{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ denotes the response vector, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$ is the design matrix, $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_n] \in \mathbb{R}^n$ is a noise vector, and $\boldsymbol{\theta}^* \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}^*\|_0 = s^*$ is the underlying sparse regression coefficient vector that we want to recover. In the high-dimensional regime, we have $n \ll d$. In order to estimate the sparse parameter vector $\boldsymbol{\theta}^*$, according to (1.1), we consider the following sparsity constrained optimization problem, which has been studied in many previous works [Zhang, 2011; Yuan *et al.*, 2014; Jain *et al.*, 2014; Chen and Gu, 2016]

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) := \frac{1}{2n}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 \text{ subject to } \|\boldsymbol{\theta}\|_0 \le s, \quad (5.2)$$

where we have each component function as $\ell_i(\boldsymbol{\theta}) = \big(\langle\mathbf{x}_i, \boldsymbol{\theta}\rangle - y_i\big)^2/2$. The next corollary provides the privacy and utility guarantees of Algorithm 1 for solving (5.2).

Corollary 5.6. Suppose each row of the design matrix $\mathbf{x}_i$ is an independent sub-Gaussian random vector with $\|\mathbf{x}_i\|_2 \le K$, and the noise vector $\boldsymbol{\xi} \sim N(0, \nu^2\mathbf{I}_n)$. For a given privacy budget $\epsilon > 0$ and a constant $\delta \in (0, 1)$, there exist constants $\{C_i\}_{i=1}^3$ such that if $n \ge C_1 s\log d$, and we choose $\sigma^2 = 2\lambda TK^2(\sqrt{2s}\|\boldsymbol{\theta}^*\|_2 + \nu\log n)^2\log(1/\delta)/(n^2\epsilon^2)$, appropriate η, large enough s, then for $T = C_2\kappa\log\big(n^2\epsilon^2/(sK^2\log d\log(1/\delta))\big)$, the output $\boldsymbol{\theta}_T$ of Algorithm 1 satisfies (ϵ, δ)-DP. In addition, with probability at least $1 - \exp(-C_3 n)$, we have

$$\mathbb{E}\|\boldsymbol{\theta}_T - \boldsymbol{\theta}^*\|_2^2 \le C_4\nu^2K^2\frac{s^*\log d}{n} + C_5K^2\big(\|\boldsymbol{\theta}^*\|_2^2 + \nu^2\big)\frac{s^{*2}\log d\log(1/\delta)}{n^2\epsilon^2},$$

where C_4, C_5 are some constants depending on log terms, which are small constants.

Remark 5.7. Corollary 5.6 implies that our algorithm achieves $O\big(s^*\log d/n + s^{*2}\log d\log(1/\delta)/(n^2\epsilon^2)\big)$ utility guarantee in the setting of (ϵ, δ)-DP. The term $O(s^*\log d/n)$ denotes the statistical error for sparse vector estimation, which matches the minimax lower bound [Raskutti *et al.*, 2011]. The term $O(s^{*2}\log d\log(1/\delta)/(n^2\epsilon^2))$ corresponds to the error introduced by the Gaussian mechanism, which matches the best known result [Kifer *et al.*, 2012].

Sparse Logistic Regression

For logistic regression, we assume that each observation y_i is drawn from the following Bernoulli distribution $\mathbb{P}(y_i = 1|\mathbf{x}_i, \boldsymbol{\theta}^*) = \exp\big(\langle\boldsymbol{\theta}^*, \mathbf{x}_i\rangle - \log\big(1+\exp(\langle\boldsymbol{\theta}^*, \mathbf{x}_i\rangle)\big)\big)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the predictive vector, $\boldsymbol{\theta}^* \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}^*\|_0 = s^*$ is the underlying parameter vector we want to recover. According to (1.1), we propose to solve the following sparsity constrained maximum likelihood estimation problem [Yuan *et al.*, 2014; Li *et al.*, 2016; Chen and Gu, 2016]

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} L_S(\boldsymbol{\theta}) := -\frac{1}{n}\sum_{i=1}^n \big[y_i\langle\boldsymbol{\theta}, \mathbf{x}_i\rangle - \log\big(1+\exp(\langle\boldsymbol{\theta}, \mathbf{x}_i\rangle)\big)\big]$$
$$\text{subject to} \quad \|\boldsymbol{\theta}\|_0 \le s, \tag{5.3}$$

where we have each component function as $\ell_i(\boldsymbol{\theta}) = \log\big(1+\exp(\langle\boldsymbol{\theta}, \mathbf{x}_i\rangle)\big) - y_i\langle\boldsymbol{\theta}, \mathbf{x}_i\rangle$. We have the following theoretical guarantees for sparse logistic regression.

Corollary 5.8. Suppose each row of the design matrix $\mathbf{x}_i$ is independent sub-Gaussian random vector and $\|\mathbf{x}_i\|_2 \le K$. For a given privacy budget $\epsilon > 0$ and a constant $\delta \in (0,1)$, there exist constants $\{C_i\}_{i=1}^4$ such that if $n \ge C_1 s\log d$, and we choose $\sigma^2 = TK^2\log(1/\delta)/(n^2\epsilon^2)$, appropriate η, large enough s, then for $T = C_2\kappa\log\big(n^2\epsilon^2/(sK^2\log d\log(1/\delta))\big)$, the output $\boldsymbol{\theta}_T$ of Algorithm 1 is (ϵ,δ)-DP. In addition, with probability at least $1-\exp(-C_3 n) - C_4/d$, we have

$$\mathbb{E}\|\boldsymbol{\theta}_T - \boldsymbol{\theta}^*\|_2^2 \le C_5 K^2\frac{s^*\log d}{n} + C_6 K^2\frac{s^*\log d}{n^2\epsilon^2}\log(1/\delta),$$

where C_5, C_6 are some constants depending on log terms, which are small constants.

Remark 5.9. In the setting of (ϵ,δ)-DP, our proposed algorithm can achieve $O(s^*\log d/n + s^*\log d\log(1/\delta)/(n^2\epsilon^2))$ utility guarantee after $T = O(\log\big(n^2\epsilon^2/s\big))$ iterations. In particular, the term $O(s^*\log d/n)$ corresponds to the statistical error, while the term $O(s^*\log d\log(1/\delta)/(n^2\epsilon^2))$ denotes the error caused by the Gaussian mechanism. To the best of our knowledge, this is the first utility guarantee for sparse logistic regression.

6 Numerical Experiments

In this section, we present experimental results of our proposed algorithm on both synthetic and real datasets. We compare our algorithm with Two stage [Kifer *et al.*, 2012] and Frank-Wolfe [Talwar *et al.*, 2015] methods. Although these two approaches were originally proposed for sparse linear regression, and have no theoretical guarantees for sparse logistic regression, they can still be applied to sparse logistic regression and produce reasonable empirical results. Thus we also include them as two baselines for sparse logistic regression. For all the experiments, we choose the variance of the random noise of different methods as suggested by their theoretical guarantees, and select other parameters, such as the step size, iteration number, and thresholding parameter by five-fold cross-validation. Note that we use the non-private iterative gradient hard thresholding method as the non-private baseline. In contrast to DP-IGHT, the non-private IGHT does not add any noise in the gradient step.

Synthetic Data Experiments

We first investigate the performances of different methods on synthetic datasets for sparse linear and logistic regression.

Sparse Linear Regression. For sparse linear regression, the underlying sparse vector $\boldsymbol{\theta}^*$ has s^* nonzero entries that are drawn independently from a uniform distribution over the interval $(-1,1)$. We generate the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ such that each element of $\mathbf{X}$ follows i.i.d. uniform distribution over the interval $(-2,2)$, then we scale each row $\mathbf{x}_i$ such that $\|\mathbf{x}_i\|_2 \le 2s^*$. The observation is generated according to $\mathbf{y} = \mathbf{X}^\top\boldsymbol{\theta}^* + \boldsymbol{\xi}$, where the noise vector $\boldsymbol{\xi} \sim N(0,\nu^2\mathbf{I})$ with $\nu^2 = 0.1$. We consider the following settings: (i) $n = d = 1000, s^* = 10$; (ii) $n = d = 5000, s^* = 30$. We set $\delta = 0.01$ and vary the privacy budget ϵ from 2 to 10. 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 1(a) and 1(b) illustrate the relative estimation error $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2/\|\boldsymbol{\theta}^*\|_2$ versus privacy budget of different methods over 10 trails. We can see that the relative estimation errors of our method are close to the non-private baseline (IGHT), and are better than existing private methods.

Sparse Logistic Regression. For sparse logistic regression, we generate the underlying sparse vector $\boldsymbol{\theta}^*$ and the design matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ in the same way as sparse linear regression. Each observation y_i is generated from the following logistic distribution

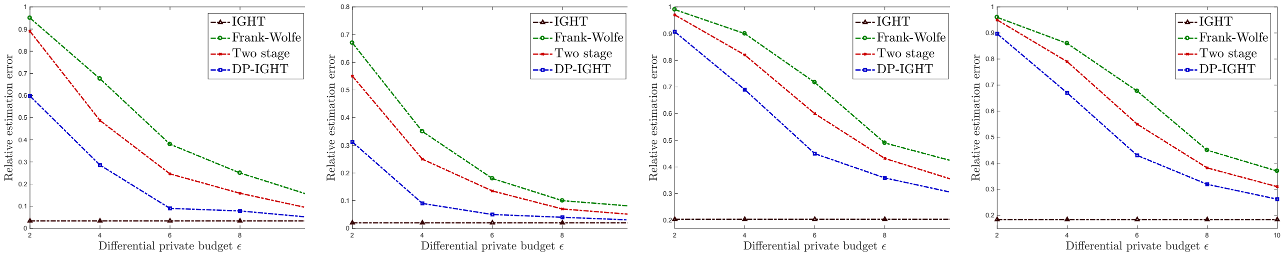
$$y_i = \begin{cases} 1, & \text{with probability } 1/\big(1+\exp(\langle\mathbf{x}_i, \boldsymbol{\theta}^*\rangle)\big), \\ 0, & \text{with probability } 1-1/\big(1+\exp(\langle\mathbf{x}_i, \boldsymbol{\theta}^*\rangle)\big). \end{cases}$$

We also consider the following two settings: (i) $n = 1000, d = 1000, s^* = 10$; (ii) $n = 5000, d = 5000, s^* = 30$. In addition, we choose the privacy budget ϵ from 2 to 10, and set $\delta = 0.01$. We demonstrate the relative estimation error $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2/\|\boldsymbol{\theta}^*\|_2$ versus privacy budget ϵ of different methods in Figure 1(c) and 1(d). The results show that our method can output accurate estimators when we have relative large privacy budget. In addition, it consistently outperforms the baseline algorithms under different privacy budget.

Real Data Experiments

In this experiment, we use two real datasets, 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 risk data from thousands of U.S. companies. In detail, it contains 16087 training examples, 3308 testing examples, and we randomly sample 50000 features for this experiment. In addition, we set $s^* = 2000$, $\delta = 0.01$, $\epsilon \in [2,10]$. Table 2 reports the mean square error (MSE) on the test data of different methods for various privacy budgets over 10 trails. In specific, MSE on the test data is defined as follows: $\|\mathbf{X}_{\text{test}}^\top\widehat{\boldsymbol{\theta}} - \mathbf{y}_{\text{test}}\|_2^2/(2n_{\text{test}})$,



(a) Sparse linear regression (b) Sparse linear regression (c) Sparse logistic regression (d) Sparse logistic regression

Figure 1: Numerical results for sparse linear regression and sparse logistic regression. (a), (b): Relative estimation error versus privacy budget for sparse linear regression; (c), (d): Relative estimation error versus privacy budget for sparse logistic regression. All the results validate the effectiveness of our algorithm DP-IGHT.

Method	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$
IGHT	0.785	0.785	0.785	0.785	0.785
Frank-Wolfe	1.514 (0.093)	1.320 (0.090)	1.210 (0.084)	1.105 (0.079)	1.094 (0.071)
Two stage	1.286 (0.112)	1.072 (0.101)	1.042 (0.082)	0.997 (0.080)	0.986 (0.075)
DP-IGHT	1.057 (0.107)	0.890 (0.081)	0.854 (0.073)	0.823 (0.070)	0.810 (0.066)

Table 2: Comparison of different algorithms for various privacy budgets ϵ in terms of MSE on the test data and its corresponding standard error (in the parenthesis) on E2006-TFIDF. Note that we set $\delta = 0.01$ in this experiment.

Method	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$
IGHT	0.0625	0.0625	0.0625	0.0625	0.0625
Frank-Wolfe	0.1271 (0.0043)	0.1034 (0.0037)	0.0938 (0.0034)	0.0852 (0.0036)	0.0807 (0.0031)
Two stage	0.1213 (0.0041)	0.0989 (0.0039)	0.0893 (0.0035)	0.0810 (0.0033)	0.0791 (0.0034)
DP-IGHT	0.1168 (0.0038)	0.0956 (0.0035)	0.0841 (0.0037)	0.0797 (0.0030)	0.0762 (0.0032)

Table 3: Comparison of different algorithms for various privacy budgets ϵ in terms of test error and its corresponding standard deviation on RCV1 data. Note that we set $\delta = 0.01$ in this experiment.

where $\{\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}}\}$ are the test data, n_{test} is the number of test examples, and $\widehat{\boldsymbol{\theta}}$ is the estimator learned on the training data. The results in Table 2 show that the performance of our algorithm is close to the non-private baseline (i.e., IGHT), and is much better than Frank-Wolfe and Two stage.

RCV1 Data. In order to compare different algorithms for sparse logistic regression, we use RCV1 dataset, which is a Reuters Corpus Volume I data set for text categorization research. More specifically, RCV1 is an archive of over 800000 manually categorized newswire stories made available by Reuters, Ltd. for research purposes. It contains 20242 training examples, 677399 testing examples and 47236 features. We use the whole training dataset and a subset of the test dataset, which contains 20000 testing examples for our experiment. In detail, we set $s^* = 500$, $\delta = 0.01$, $\epsilon \in [2, 10]$. We compare all algorithms in terms of their classification error on the test set over 10 replications, which is summarized in Table 3. It is obvious that our algorithm achieves the lowest test error among private algorithms on RCV1 dataset, which demonstrates the superiority of our algorithm.

7 Conclusions

In this paper, we proposed a privacy preserving iterative gradient hard thresholding algorithm for sparse learning. We establish a linear convergence rate and strong utility guarantee of our algorithm. Experiments on both synthetic and real world data demonstrate the superiority of our algorithm.

Acknowledgements

This research was sponsored in part by the National Science Foundation SaTC-1717950.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[Bassily *et al.*, 2014] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. *arXiv preprint arXiv:1405.7085*, 2014.

[Bickel *et al.*, 2009] Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[Bun and Steinke, 2016] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[Chaudhuri and Monteleoni, 2009] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression.

In *Advances in Neural Information Processing Systems*, pages 289–296, 2009.

[Chaudhuri *et al.*, 2011] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[Chen and Gu, 2016] Jinghui Chen and Quanquan Gu. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *UAI*, 2016.

[Ding *et al.*, 2017] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*, pages 3571–3580, 2017.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[Erlingsson *et al.*, 2014] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[Jain and Thakurta, 2014] Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484, 2014.

[Jain *et al.*, 2014] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014.

[Jayaraman *et al.*, 2018] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 6346–6357, 2018.

[Kifer *et al.*, 2012] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1, 2012.

[Kogan *et al.*, 2009] Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. 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*, pages 272–280. Association for Computational Linguistics, 2009.

[Lewis *et al.*, 2004] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[Li *et al.*, 2016] Xingguo Li, Raman Arora, Han Liu, Jarvis Haupt, and Tuo Zhao. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*, 2016.

[Loh and Wainwright, 2013] Po-Ling Loh and Martin J Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.

[Negahban *et al.*, 2009] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[Raskutti *et al.*, 2011] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.

[Rényi, 1961] Alfréd Rényi. On measures of entropy and information. Technical report, HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, 1961.

[Talwar *et al.*, 2015] Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3025–3033, 2015.

[Thakurta and Smith, 2013] Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850, 2013.

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

[Van de Geer and others, 2008] Sara A Van de Geer et al. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2):614–645, 2008.

[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 2719–2728, 2017.

[Wang *et al.*, 2018] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. In *Advances in Neural Information Processing Systems*, pages 973–982, 2018.

[Xu *et al.*, 2012] Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):187–193, 2012.

[Yuan *et al.*, 2014] Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning*, pages 127–135, 2014.

[Zhang *et al.*, 2017] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*, 2017.

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