Stochastic Privacy-Preserving Methods for Nonconvex Sparse Learning

Guannan Liang[a†], Qianqian Tong[a,†], Jiahao Ding[b], Miao Pan[b], Jinbo Bi[a*]

[a] *University of Connecticut,* [b] *University of Houston*

Abstract

Sparse learning is essential in mining high-dimensional data. Iterative hard thresholding (IHT) methods are effective for optimizing nonconvex objectives for sparse learning. However, IHT methods are vulnerable to adversary attacks that infer sensitive data. Although pioneering works attempted to relieve such vulnerability, they confront the issue of high computational cost for large-scale problems. We propose two differentially private stochastic IHT: one based on the stochastic gradient descent method (DP-SGD-HT) and the other based on the stochastically controlled stochastic gradient method (DP-SCSG-HT). The DP-SGD-HT method perturbs stochastic gradients with small Gaussian noise rather than full gradients, which are computationally expensive. As a result, computational complexity is reduced from $O(n\log(n))$ to a lower $O(b\log(n))$, where n is the sample size and b is the mini-batch size used to compute stochastic gradients. The DP-SCSG-HT method further perturbs the stochastic gradients controlled by large-batch snapshot gradients to reduce stochastic gradient variance. We prove that both algorithms guarantee differential privacy and have linear convergence rates with estimation bias. A utility analysis examines the relationship between convergence rate and the level of perturbation, yielding the best-known utility bound for nonconvex sparse optimization. Extensive experiments show that our algorithms outperform existing methods.

Keywords: Sparse learning, differential privacy, stochastic algorithm

1. Introduction

Sparse learning decreases the data dimension effectively in predictive modeling. It plays an important role in various data mining fields, including bioinformatics, image analysis, and engineering. Numerous successful sparse learning applications for high-dimensional problems depend on the cardinality constraint for sparsity which poses difficulties for the statistical and computational analysis of such an approach. In this study, we investigate the following cardinality-constrained nonconvex empirical risk minimization (ERM) problem:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{z=1}^{n} f_z(x) \quad \text{subject to} \quad \|x\|_0 \le k, \tag{1}$$

where $f(x)$ is a smooth function, $f_z(x)$ ($z \in [n] := \{1, 2, \ldots, n\}$) is an individual loss associated with the z^{th} sample, $\|x\|_0$ denotes the l_0-norm of the vector x which computes the number of nonzero entries in x, and the integer k specifies the required level of sparsity. Problem (1) appears in many statistical learning, machine learning, and signal processing problems and is widely used in high-dimensional data analyses. Because the cardinality constraint, $\|x\|_0 \le k$, is nonconvex, Problem (1) is a nonconvex constrained optimization problem, and finding a global optimal solution x^* to Problem (1) is generally NP-hard.

Existing research for Problem (1) primarily falls within the regimes of either iterative hard thresholding (HT) methods [6, 18] or matching pursuit methods [30, 34]. Even though matching pursuit methods achieve remarkable

[†] These authors contributed equally to this work.

[*] Corresponding author.

Email addresses: guannan.liang@uconn.edu (Guannan Liang[a]), qianqian.tong@uconn.edu (Qianqian Tong[a,†]), jding7@uh.edu (Jiahao Ding[b]), mpan2@uh.edu (Miao Pan[b]), jinbo.bi@uconn.edu (Jinbo Bi[a])

Preprint submitted to Information Sciences *November 1, 2022*

Table 1: Comparison of our approaches against the existing DP-GD-HT that is based on the regular gradient descent method. The DP-SGD-HT computes stochastic gradients based on mini-batches of size b which is $\ll n$ in practice, so it has less computational complexity than the DP-GD-HT (see the last column). A necessary assumption used to prove the convergence of the DP-SGD-HT is that the variance of stochastic gradients is upper bounded by σ_0^2. This assumption is no longer enforced when full gradients are used to correct for the variance of stochastic gradients as in the DP-SVRG-HT. The SCSG uses gradients computed on large data batches (size B) to correct for the variance of mini-batch-based gradients. The parameter $\psi \ll 1$ is defined in Corollary 5.5.1, so the DP-SCSG-HT also has smaller complexity than the DP-GD-HT.

Algorithm	Reference	Full Gradient	Constraint on variance σ_0^2	Computational Complexity
DP-GD-HT	[42]	Yes	No	$O(n \log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$
DP-SGD-HT	This work	No	Yes	$O(b \log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$
DP-SVRG-HT[1]	This work	Yes	No	$O(n \log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$
DP-SCSG-HT	This work	No	Yes	$O(\min\{1, \psi\} \cdot n \log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$

[1] A special case of DP-SCSG-HT, with batch size $B = n$.

success in quadratic loss functions (e.g., the l_0-constrained linear regression problems), they are required to find an optimal solution to min $f(x)$ on the identified support. The support is defined as the entries of x that are non-zero after hard thresholding. This minimization problem has no analytical solution for arbitrary (non-quadratic) losses, making its solution time-consuming [4]. Thus, iterative gradient-based HT methods have become popular for nonconvex sparse learning.

In many sparse learning applications, the data is highly sensitive, e.g., genomic data, financial and electronic medical records. Without safeguards to protect privacy, adversaries may attack the deployed model in an attempt to infer private information via a membership inference attack or through feature leakage. Therefore, ensuring sensitive information is adequately protected from malicious parties is of critical importance. To address this, the machine learning and deep learning communities have developed algorithms with differential privacy (DP) for unconstrained optimization problems. These extensively studied methods include three common approaches: output perturbation [46], objective perturbation [7], and gradient perturbation [5, 2, 40]. Conceptually, in the output perturbation mechanism, the learning algorithm runs the same as in its non-DP case, and then noise is added to the output parameter. The objective perturbation includes a noise term to the objective function which is the empirical loss, then releases the minimizer of the perturbed objective. In contrast, the gradient perturbation is to inject noise at every iterative to gradient updates. While these methods provide solutions when the problem is unconstrained, privacy-preserving guarantees in the sparse learning setting have been under-explored, especially in the context of stochastic optimization.

Several studies attempt to develop differentially private algorithms for sparse learning problems, such as the least absolute shrinkage and selection operator (Lasso) problem [23, 36] or the cardinality constrained problem [42]. The Lasso problem uses the l_1-norm, i.e., $\|x\|_1 = \sum |x_j|$ to regularize the model parameters x whereas the cardinality constrained problem uses the l_0-norm which counts the non-zero entries in the parameter vector x. The l_1-norm is a convex surrogate of the l_0-norm. Assuming a convex loss function is used [23, 36], the Lasso problem is a convex relaxation of Problem (1) and is easily solved by gradient-based methods. However, this can result in large estimation bias in the solution to Problem (1), and has been shown to have worse empirical performance [27]. Recent research has focused directly on cardinality constrained problems, producing a differentially private, gradient based algorithm which utilizes HT (DP-GD-HT) [42]. Although the DP-GD-HT algorithm has a competitive utility analysis, which investigates the trade-off between the convergence rate and level of perturbation, it is not a stochastic algorithm. As DP-GD-HT requires computing the full gradient at each iteration, it is computationally expensive for high-dimensional problems with large sample sizes.

In this paper, we propose and analyze two differentially private algorithms, DP-SGD-HT and DP-SCSG-HT, to solve Problem (1). We prove their convergence rates, utility bounds, and computational complexities. Using benchmark financial and medical records, we also conduct an experimental validation of our theoretical analysis for the proposed stochastic privacy-preserving methods. Our contributions are as follows.

- We design the first differentially private stochastic iterative HT method (DP-SGD-HT) that reduces the computational cost while guaranteeing the DP. Then, to reduce the variance of stochastic gradients and further improve learning accuracy, we develop a second DP algorithm called the Stochastically Controlled Stochastic Gradient HT method (DP-SCSG-HT). However, the privacy analysis of DP-SCSG-HT is difficult due to the random number of iterations per epoch. We provide a refined and precise estimation of privacy loss for DP-SCSG-HT using RDP by controlling the effect of random iteration numbers.

- We prove that the sequence $\{x_0, x_1, \cdots, x_T\}$ generated by either DP-SGD-HT or DP-SCSG-HT satisfies $E[\|x_T - x^*\|^2] \leq \theta^T \|x_0 - x^*\|^2 + e$, where $0 < \theta < 1$ and e is the statistical bias due to the sparsity requirement and the injected Gaussian noise. It means that the two algorithms both enjoy a linear convergence rate with a linear factor θ under a statistical bias e, and that their results match those of non-stochastic DP-GD-HT, which also converges at a linear rate. Despite the stochastic manner of the proposed algorithms, we prove their utility is preserved, matching the best-known utility bound obtained in the DP-GD-HT [42].
- We demonstrate that stochastic methods significantly lower the computational complexity of the DP-GD-HT, as shown in Table 1. The complexity of the DP-SGD-HT is linearly dependent on $O(b\log(n))$ as compared to $O(n\log(n))$ for the DP-GD-HT, where b is the size of the mini-batch used to compute the stochastic gradients and can be significantly less than n. The computational complexity of the DP-SCSG-HT is $O(\min\{1, \psi\} \cdot n\log(n))$, where $\psi \ll 1$ in practice.

The sections of this paper are organized as follows. The second section introduces the related work. The third section gives preliminaries, including notations, definitions, lemmas and assumptions. Furthermore, Sections 4 and 5 provide a detailed study, privacy analysis, convergence analysis and utility analysis of the proposed methods. Section 6 introduces the experimental results and performance analyses. Finally, in Section 7, we present the conclusion.

2. Related work

2.1. Differential privacy

To protect user privacy, a learning algorithm/mechanism can be designed to satisfy the (ϵ, δ)-Differential Privacy (DP), which is a widely adopted mathematical definition of privacy-preserving and has become a standard in academic and industrial fields due to its provable protection against adversaries [15, 32, 28, 19, 16, 41]. The formal definition of DP is as follows:

Definition 1 ((ϵ, δ)-DP [10])**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies the (ϵ, δ)-differential privacy ((ϵ, δ)-DP) if for any two adjacent datasets $D, D' \in \mathcal{D}$, for any output set $O \subseteq \mathcal{R}$, it holds that $\mathbb{P}[\mathcal{M}(D) \in O] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(D') \in O] + \delta$, where $\mathcal{R}$ is the output space of $\mathcal{M}$ and the adjacent sets mean that D and D' differ by one entry.*

When sharing individuals' data with other organizations or the public, the (ϵ, δ)-DP is a measurement of how much privacy one can withhold for an individual whose data is included in the dataset. Mathematically, the (ϵ, δ)-DP implies that the mechanism $\mathcal{M}$ is ϵ-indistinguishable between two adjacent sets with probability $1 - \delta$. For any output set O, $\frac{\mathbb{P}[\mathcal{M}(D) \in O]}{\mathbb{P}[\mathcal{M}(D') \in O]} \in [e^{-\epsilon}, e^{\epsilon}]$ with high probability $1 - \delta$ and particularly, when ϵ is close to 0, $e^{\epsilon} \approx 1 + \epsilon$, so $\frac{\mathbb{P}[\mathcal{M}(D) \in O]}{\mathbb{P}[\mathcal{M}(D') \in O]} \in [1 - \epsilon, 1 + \epsilon]$.

Definition 1 of DP contributes to data privacy protection. For example, in the membership inference attack against a machine learning model published on cloud platforms such as Amazon [8] and IBM [48], an attacker may be able to infer, based on the model prediction of an example z, whether z belongs to the training data on which the model was trained. In this example, consider that a cancer treatment center has trained and provided z with a machine learning model, and that z is a patient of the center. If a non-DP approach, such as stochastic gradient descent, is used to train the model, his or her PHI (i.e. cancer diagnosis) may be disclosed. However, if the model is trained using a DP technique, regardless of whether or not z belongs to the training data D (assuming the D' differs from D by just z), the model will have limited variation (by e^{ϵ} as defined in Definition 1) for the attacker to detect the membership [38].

The parameter ϵ is commonly referred to as privacy budget and δ is considered as the exceptional probability. In other words, with the probability δ, the model may vary beyond e^{ϵ} when training on adjacent training datasets. DP, on the other hand, is equivalent to setting a constraint on the model so that it learns more from the training data as a whole than from a single training example. Consequently, DP frequently results in a decline in prediction accuracy. A smaller (restricted) privacy budget ϵ corresponds to lower prediction accuracy.

2.2. *Optimization methods*

For the unconstrained ERM problem of

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{z=1}^{n} f_z(x), \tag{2}$$

the stochastic gradient descent (SGD) method and its variants – variance reduced methods, such as the stochastic variance reduced gradient (SVRG) [20] and stochastically controlled stochastic gradient (SCSG) [24] methods – have been extensively studied. However, these methods are proposed for unconstrained optimization and are not directly applicable to cardinality-constrained Problem (1). Due to the non-convexity of the cardinality constraint, Problem (1) is difficult to solve even without the privacy-preserving concern.

Iterative gradient-based HT methods, such as gradient descent HT (GD-HT) [18], stochastic gradient descent HT (SGD-HT) [33], hybrid stochastic gradient HT (HSG-HT) [47], stochastic variance reduced gradient HT (SVRG-HT) [27], and stochastically controlled stochastic gradient HT (SCSG-HT) [29] have emerged as the dominant force in nonconvex sparse learning. These techniques use gradient descent or one of its variants to update the iterate x^t before using the HT operator to enforce the x^t's sparsity. The computation can be concisely written as $x^{t+1} = \mathcal{H}_k(x^t - \eta v^t)$, where η is the learning rate, v^t can be the full gradient, stochastic gradient or variance reduced gradient at the t^{th} iteration, and $\mathcal{H}_k(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ denotes the HT operator that preserves the largest k elements of x in magnitude and sets other elements to 0. In a distributed computing setting, these iterative HT algorithms share gradients computed on a local device to other devices or a central server, and the shared gradients may leak private data when training a machine learning model.

3. Preliminaries

Notations. We denote a vector by a lowercase letter, e.g. x, and the l_0-norm and l_2-norm of vector x by $\|x\|_0$ and $\|x\| = \sum x_j^2$ respectively. For any vectors $a, b \in \mathbb{R}^d$, we use $\langle a, b \rangle$ to denote the inner product of a and b. An identity matrix is denoted by $\mathbf{I}$. Let $O(\cdot)$, $\Omega(\cdot)$ and $\Theta(\cdot)$ represent the asymptotic upper, lower, and tight bounds, respectively, and $E[\cdot]$ represent taking expectation over all random variables. We denote the integer set $\{1, ..., n\}$ by $[n]$, and $\nabla f(\cdot)$, $\nabla f_I(\cdot)$ and $\nabla f_z(\cdot)$ are the full gradient, stochastic gradient over a mini-batch $I \subset [n]$, and stochastic gradient over a training example indexed by $z \in [n]$, respectively. The symbol $\mathbb{I}(\cdot)$ is an indicator function, and $supp(x)$ means the support of x or the index set of non-zero elements in x. Let x^* be the optimal solution of Problem (1). The support $\mathcal{I}_{t+1}^{(j)} = supp(x^*) \cup supp(x_t^{(j)}) \cup supp(x_{t+1}^{(j)})$, is associated with the $(t+1)$-th iteration at the j-th epoch (and $\mathcal{I}$ is used throughout the paper without ambiguity); $\widetilde{\mathcal{I}} = supp\left(\mathcal{H}_{2k}\left(\nabla f\left(\boldsymbol{x}^*\right)\right)\right) \cup supp\left(\boldsymbol{x}^*\right)$. The projector $\pi_{\mathcal{I}}(x)$ gives a vector of the same length as x but zeros out the elements of x not indexed in $\mathcal{I}$. All parameters used in our analysis are listed in Table 2 with parameter constraints for easy access.

3.1. *Rényi differential privacy*

To measure the distance between the model output distributions from two adjacent datasets, we introduce Rényi Divergence as a measure for distributions, which generalizes the Kullback-Leibler (KL) divergence with a parameter α.

Definition 2 (Rényi Divergence [35])**.** *Let P and Q be probability distributions on Ω. For $\alpha \in (1, \infty)$, the Rényi Divergence of order α between P and Q is defined as* $\mathrm{D}_\alpha(P \| Q) = \frac{1}{\alpha - 1} \log \left(\int_\Omega P(x)^\alpha Q(x)^{1-\alpha} \mathrm{d}x \right)$.

Even though (ϵ, δ)-DP has been a commonly used concept in privacy preserving research communities, it can be challenging to apply it to the study of iterative algorithms due to the randomized mechanism's recursive repeating in the iterations, which necessitates rule of composition. The (α, ρ)-Rényi differential privacy (RDP) is a generalization of the (ϵ, δ)-DP that makes it simpler to read and combine rules over iterations. Our theoretical analyses are therefore based on the (α, ρ)-RDP.

Table 2: Definitions and constraints of the parameters used in our algorithms and analysis.

Notation	Definition	Constraint
z	data sample	$z \in [n]$
k	number of nonzero entries	
$\mathcal{H}_k(\cdot)$	hard-thresholding operator	
ϵ	privacy budget	
δ	exceptional probability for $(\epsilon, 0)$-DP	
(α, ρ)	Rényi differential privacy	(α, ρ)-RDP equals to $(\rho + \frac{log(1/\delta)}{\alpha-1}, \delta)$-DP
S, S'	adjacent datasets	S, S' differs by one example
$\Delta_2(q)$	l_2-sensitivity for query q	$\Delta_2(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$
$\mathbf{I}$	identity matrix	
ρ_s	restricted strongly convex	
L_s	restricted strongly smooth	
t	iteration index for Alg.1 or inner loop index for Alg.2	
j	outer loop index for Alg.2	
$\mathcal{T}$	total number of iterations for Alg.1	$T = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$
$\mathcal{J}$	total number of outer loop for Alg.2	$\mathcal{J} = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$
B/b	number of inner loop's iteration in Alg. 2	$B/b = \Theta(\sqrt{k})$
$e^{(j)}$	bias of $v_t^{(j)}$	$e^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)}) - \nabla f(\tilde{x}^{(j)})$
κ_s	restricted condition number	$\kappa_s = \frac{L_s}{\rho_s}$
$\mathcal{I}$ or $\mathcal{I}_{t+1}^{(j)}$	Support	$\mathcal{I} = supp(x^*) \cup supp(x_t^{(j)}) \cup supp(x_{t+1}^{(j)})$
$\tilde{\mathcal{I}}$	Support	$\tilde{\mathcal{I}} = supp(x^*) \cup supp(\mathcal{H}_{2k}(f(x^*)))$
$\pi_{\mathcal{I}}(\cdot)$	projection on support $\mathcal{I}$	
β	parameter in truncation lemma	$\beta = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}}$

Definition 3 ((α,ρ)-RDP [31]). *We call a randomized mechanism $\mathcal{M} : \mathcal{S} \rightarrow \mathcal{R}$ satisfies (α,ρ)-Rényi differential privacy, or shorten as (α,ρ)-RDP, if for any two adjacent datasets $S, S' \in \mathcal{S}$, (i.e., differing by one example), the inequality $D_\alpha(\mathcal{M}(S)\|\mathcal{M}(S')) \leq \rho$ holds for $\alpha \in (1,\infty)$ and $\rho \in (0,\infty)$, where $D_\alpha(\mathcal{M}(S)\|\mathcal{M}(S'))$ is the α-Rényi divergence between two distributions $\mathcal{M}(S)$ and $\mathcal{M}(S')$, $\mathcal{S}$ is the space containing all possible sample sets that have n samples from an underlying distribution.*

Intuitively, to achieve DP, the algorithm needs to alleviate the influence of any single data point on the final model in such a way that the model derived from a training dataset that only lacks a single data example makes very similar predictions to the model trained with that data example included. Formally, the magnitude with which a single data point can alter the final model should be measured in the worst-case scenario. As a result, an algorithm's sensitivity is introduced to provide an upper bound on how much perturbation can be tolerated while still maintaining privacy.

Definition 4 (l_2-sensitivity [11]). *For any two adjacent datasets $S, S' \in \mathcal{S}$, the l_2-sensitivity $\Delta_2(q)$ of a query $q : \mathcal{S} \rightarrow \mathcal{R}$ is defined as $\Delta_2(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$ where* sup *means taking the superior of the l_2-norm over all possible pairs of adjacent datasets.*

Remark 3.1. *In recent studies, the (α,ρ)-RDP has been used as an alternative of the (ϵ,δ)-DP. The (α,ρ)-RDP corresponds to the $(\rho + \frac{log(1/\delta)}{\alpha-1}, \delta)$-DP for any $\delta \in (0,1)$, which allows us to convert from (α,ρ)-RDP to (ϵ,δ)-DP.*

In machine learning, the widely used SGD algorithm subsamples mini-batches from the training dataset S. In a distributed computing environment, communicating stochastic gradients based on mini-batches rather than full gradients helps to protect data privacy. However, it imposes challenges to the traditional DP analysis based on the l_2-sensitivity, which is defined over the whole dataset S. Recently, the privacy amplification theorem for DP [21] shows that if $\mathcal{M}$ is (ϵ,δ)-DP, then $\mathcal{M}$ with the subsampling mechanism is $(O(\tau\epsilon), \tau\delta)$-DP where τ is the subsampling rate. We prefer the (α,ρ)-RDP because, as demonstrated in Lemma 3.2 [45, 44], RDP has an analytical and tighter bound for subsampling mechanism. Using the definition of the l_2-sensitivity, the following lemmas have been proved for the Gaussian mechanism and the composition rule of RDP.

Lemma 3.2 (Gaussian mechanism [44]). *Given a function $q : \mathcal{S} \rightarrow \mathcal{R}$, and $u \sim N(0, \sigma^2\mathbf{I})$, the Gaussian mechanism $\mathcal{M} = q(S) + u$ satisfies $(\alpha, \frac{\alpha\Delta_2^2(q)}{2\sigma^2})$-RDP. If we apply $\mathcal{M}$ to subsamples that are uniformly sampled without replacement from S, $\mathcal{M}$ satisfies $(\alpha, \frac{5\tau^2\alpha\Delta_2^2(q)}{\sigma^2})$-RDP, if $\alpha \leq \log(\frac{1}{\tau(1+\sigma^2/\Delta_2^2(q))})$, where $\sigma^2 \geq 1.5\Delta_2^2(q)$ and τ is the subsampling rate.*

The privacy amplification theorem for RDP in [44] proves that the Gaussian perturbation parameter σ^2 needs to satisfy $\sigma^2 \geq 1.5\Delta_2^2(q)$ in order to derive an analytical formulation for ρ (in Lemma 3.2). It means that $\alpha \leq \log(\frac{1}{\tau(1+\sigma^2/\Delta_2^2(q))}) \leq -\log(2.5\tau)$. From Definition 3, $\alpha > 1$ is required in the (α,ρ)-RDP, hence the sampling rate τ satisfies that $\tau < e^{-1}/2.5 \approx 0.147$.

Lemma 3.3 (RDP composition [31]). *For two randomized mechanisms $\mathcal{M}_1 : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{R}$ and $\mathcal{M}_2 : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{R}$, if $\mathcal{M}_1$ satisfies (α,ρ_1)-RDP and $\mathcal{M}_2$ satisfies (α,ρ_2)-RDP, then the process of $\mathcal{M}_2(S, \mathcal{M}_1(S))$ (as a joint random process with $\mathcal{M}_1(S,\cdot)$) satisfies $(\alpha, \rho_1 + \rho_2)$-RDP.*

In this paper, a mechanism corresponds to a single SGD iteration with injected Gaussian noise. If our algorithm runs T iterations in total, we can recursively use Lemma 3.3 , so if the t-th mechanism satisfies (α,ρ_t)-RDP, then the composition of T mechanisms brings the entire algorithm to be $(\alpha, \sum_{t=1}^{T}\rho_t)$-RDP.

Lemma 3.4 (Invariant of post-processing [31]). *For mechanism $\mathcal{M}$ and post-processing mapping $g : \mathcal{R} \rightarrow \mathcal{R}$, if $\mathcal{M}$ satisfies (α,ρ)-RDP, then $g(\mathcal{M}(\cdot))$ is still (α,ρ)-RDP.*

3.2. Assumptions

Throughout the theoretical analyses, we assume that the objective function $f(x)$ in Problem (1) satisfies the following commonly used assumptions in the study of nonconvex optimization.

Assumption 1. *Assume that the function $f_z(x)$ is l-Lipschitz continuous for any $z \in \{1, \cdots, n\}$. In other words, there exists a constant $l \geq 0$ such that $|f_z(x_1) - f_z(x_2)| \leq l\|x_1 - x_2\|, \forall x_1, x_2 \in \mathbb{R}^d, z \in [n]$.*

Remark 3.5. *For a differentiable function, the l-Lipschitz continuity implies that the gradient of the function is upper bounded, i.e.,* $\forall x$, $\|\nabla f_z(x)\| \le l$. *Assumption 1 is commonly used for deriving the l_2-sensitivity, such as in [40, 42]. In practice, instead of assuming the Lipschitz continuity of f_z, the gradient clipping technique in [2] can be used to ensure $\|\nabla f_z(x)\|$ is upper bounded by a pre-difined value l.*

Assumption 2. *Assume that the function $f(x)$ has σ_0^2-bounded stochastic gradient variance, i.e.,* $E[\|\nabla f_z(x) - \nabla f(x)\|^2] \le \sigma_0^2, \forall x \in \mathbb{R}^d, z \in [n]$.

For fair comparison with prior works on the HT methods [18, 33, 47, 27], we also use the same assumption as follows.

Assumption 3. *Assume that the function $f(x)$ is:*

(i) restricted ρ_s-strongly convex at the sparsity level s for a given $s \in \mathbb{N}_+$, i.e., there exists a constant $\rho_s > 0$ such that $\forall x_1, x_2 \in \mathbb{R}^d$ that $\|x_1 - x_2\|_0 \le s$, we have $f(x_1) - f(x_2) - \langle \nabla f(x_2), x_1 - x_2 \rangle \ge \frac{\rho_s}{2}\|x_1 - x_2\|^2$;

(ii) restricted L_s smooth at the sparsity level s for a given $s \in \mathbb{N}_+$, i.e., there exists a constant $L_s > 0$ such that $\forall x_1, x_2 \in \mathbb{R}^d$ with $\|x_1 - x_2\|_0 \le s$, we have $f(x_1) - f(x_2) - \langle \nabla f(x_2), x_1 - x_2 \rangle \le \frac{L_s}{2}\|x_1 - x_2\|^2$.

4. The DP-SGD-HT

In this section, we propose a stochastic version of the DP HT algorithm to reduce the computation of full gradients. We name the SGD-based HT algorithm DP-SGD-HT, as shown in Algorithm 1, because it can solve the sparsity constrained optimization problem Eq.(1) in a stochastic privacy-preserving manner.

Algorithm 1 DP-SGD-HT

```
Input: The maximal number of iterations $T$, initial state $x^0$, stepsize $\eta$, the mini-batch size $\{b_t\}$ at the $t$-th iteration, privacy parameters $\epsilon$, $\delta$ and $\alpha$
for $t = 1, 2, ..T$ do
    Sample uniformly a batch of examples, $I_t \subset \{1, ..., n\}$, where $|I_t| = b_t$
    $g_t = \nabla f_{I_t}(x_t)$
    $u_t \sim N(0, \sigma^2 \mathbf{I})$ where $\sigma^2 = \frac{40\alpha l^2 T}{n^2\epsilon}$
    $x_{t+1} = \mathcal{H}_k(x_t - \eta(g_t + u_t))$
end for
```

At the core of Algorithm 1 is a stochastic gradient perturbation procedure at each iteration. Specifically, we perturb the stochastic gradient in an iteration with Gaussian noise $N(0, \sigma^2\mathbf{I})$, instead of perturbing computationally-expensive full gradients used in DP-GD-HT algorithms [42, 43, 39]. We then make use of the composition rule and privacy-amplification by subsampling of DP to prove an upper bound on the total privacy loss. Note that the DP-SGD-HT is a special case of the original SGD-HT if the noise variance $\sigma^2 = 0$, though we provide a suggested value of σ^2 in Algorithm 1. In the following, we provide the privacy analysis, convergence guarantee, and utility bound of the proposed DP-SGD-HT algorithm.

4.1. Differential Privacy Guarantee of the DP-SGD-HT

We show that Algorithm 1 satisfies the DP. Specifically, we prove that it satisfies the (α, ρ)-RDP, and then we convert it to the format of (ϵ, δ)-DP as discussed in Remark 3.1, so that it may be compared with previously published results..

Theorem 4.1. *Algorithm 1 satisfies the (ϵ, δ)-DP, when $b_t = b$, and $\sigma^2 = \frac{40\alpha l^2 T}{n^2\epsilon}$, where $\alpha = 1 + \frac{2\log(1/\delta)}{\epsilon}$, and if $\alpha \le \log(\frac{n^3\epsilon}{n^2 b\epsilon + 10\alpha T b^3})$ and $\frac{10b^2\alpha T}{n^2\epsilon}$* ≥ 1.5.

Proof. At the $(t+1)$-th iteration of Algorithm 1, we have the update rule: $x_{t+1} = \mathcal{H}_k(x_t - \eta(g_t + u_t))$, where $g_t = \nabla f_{I_t}(x_t)$ and $u_t \sim N(0, \sigma^2 \mathbf{I})$.

We consider the following query function on a set S of n training examples, $q_t(S) = \frac{1}{b_t}\sum_{i=1}^{n} \nabla f_i(x_t)$. For any two adjacent datasets S and S', let us index the different examples in S and S' by z and z'. By Definition 4 and Remark 3.5, the l_2-sensitivity $\Delta_2(q_t)$ of q_t is:

$$\Delta_2(q_t) = \sup_{S,S'} \|q_t(S) - q_t(S')\| = \sup_{z,z'} \|\frac{1}{b_t}\nabla f_z(x_t) - \frac{1}{b_t}\nabla f_{z'}(x_t)\| \leq \frac{2l}{b_t}.$$

By Lemma 3.2, the Gaussian mechanism $\mathcal{M} = q_t(S) + u_t$ is $(\alpha, \frac{2\alpha l^2}{b_t^2\sigma^2})$-RDP for the query function $q_t(S)$. We now consider $\tilde{q}_t(S)$ calculated on a subsample I_t that is uniformly drawn from S, $\tilde{q}_t(S) = \frac{1}{b_t}\sum_{z\in I_t} \nabla f_z(x_t)$. Because the sampling rate $\tau = \frac{b_t}{n}$, substituting the formula of τ and $\Delta_2(q_t)$ into Lemma 3.2 yields that $\tilde{\mathcal{M}} = \tilde{q}_t(S) + u_t$ is $(\alpha, \frac{20\alpha l^2}{n^2\sigma^2})$-RDP, if $\alpha \leq \log\left(\frac{n}{b_t(1+\frac{\sigma^2 b_t^2}{4l^2})}\right)$ and $\sigma^2 \geq \frac{6l^2}{b_t^2}$. Due to the invariant property of post-processing of RDP [31], we know that the mechanism $\tilde{\mathcal{M}}' = \mathcal{H}_k(x_t - \eta\tilde{\mathcal{M}})$ is $(\alpha, \frac{20\alpha l^2}{n^2\sigma^2})$-RDP. By Lemma 3.3, after running T iterations, we obtain that Algorithm 1 satisfies the $(\alpha, \frac{20\alpha l^2 T}{n^2\sigma^2})$-RDP, and correspondingly $(\frac{20\alpha l^2 T}{n^2\sigma^2} + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP for $\delta \in (0,1)$ according to Remark 3.1. Let

$$\frac{20\alpha l^2 T}{n^2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1} = \epsilon,$$

and $\alpha = 1 + \frac{2\log(1/\delta)}{\epsilon}$, which implies that $\sigma^2 = \frac{40\alpha l^2 T}{n^2\epsilon}$. This σ^2 formula gives us the suggested value for the injected Gaussian noise.

Therefore, Algorithm 1 satisfies the (ϵ,δ)−DP if we use $b_t = b$, $\alpha = 1 + \frac{2\log(1/\delta)}{\epsilon}$, and $\sigma^2 = \frac{40\alpha l^2 T}{n^2\epsilon}$ in Algorithm 1, and if $\alpha \leq \log(\frac{n^3\epsilon}{n^2 b\epsilon + 10\alpha T b^3})$ and $\frac{10b^2\alpha T}{n^2\epsilon} \geq 1.5$. □

Theorem 4.1 guarantees that the DP-SGD-HT algorithm is (ϵ,δ)−DP, and derives an analytical formula for σ^2, the added Gaussian noise parameter. A constraint on α is introduced as a result of the subsampling technique used in Algorithm 1. This constraint is similar to the constraint introduced in [2] for deep learning applications with the moments accountant technique, while our α has a closed-form solution. If we directly work on (ϵ,δ)-DP and apply the strong composition theorem in [12], we can remove this constraint, but an extra $\log(T/\delta)$ factor will be introduced to σ^2 and hence will worsen the utility bound derived in a later section.

4.2. Convergence Guarantee of the DP-SGD-HT

In order to make the SGD-HT satisfy the (ϵ,δ)-DP, Algorithm 1 has included a randomized Gaussian process, which may influence the convergence of the original SGD-HT technique and the convergence rate. We examine the convergence of the DP-SGD-HT by developing an upper bound on the distance between the estimator x_t and the optimal x^*, i.e. $E[\|x_t - x^*\|^2]$ in Theorem 4.2.

Theorem 4.2. *Suppose that $f(x)$ satisfies Assumptions 1 - 3, $k^* = \|x^*\|_0$, $k \geq 4k^*(12\kappa_s - 1)^2 + k^*$ where $\kappa_s = \frac{L_s}{\rho_s}$ is the condition number of $f(x)$. Define $\tilde{\mathcal{I}} = supp(x^*) \cup supp(\mathcal{H}_{2k}(\nabla f(x^*)))$, and let $\eta = \frac{1}{6L_s}$. If the variance of stochastic gradients $\sigma_0^2 \leq kb_t\sigma^2$, then we can get*

$$E[\|x_t - x^*\|^2] \leq \theta_1^t \|x_0 - x^*\|^2 + \frac{1}{1-\theta_1}\frac{1+\beta}{12L_s^2}\left\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\right\|^2 + \frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\sigma^2, \tag{3}$$

where $\beta = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}}$, $\theta_1 = (1 + \frac{2\sqrt{k^*}}{\sqrt{k-k^*}})(1 - \frac{1}{12\kappa_s}) < 1$.

Proof. Assume that $y_t = x_t - \eta(\pi_{\mathcal{I}}(g_t + u_t))$, then

$$
\begin{aligned}
E[\,\|y_t - x^*\|^2] &= E[\|x_t - \eta(\pi_{\mathcal{I}}(g_t + u_t)) - x^*\|^2] \\
&\overset{①}{=} E[\|x_t - x^*\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(g_t)\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(u_t)\|^2] - 2\eta E[\langle x_t - x^*, \pi_{\mathcal{I}}(g_t)\rangle] \\
&\overset{②}{\leq} E[\|x_t - x^*\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(g_t)\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(u_t)\|^2] - 2\eta E[f(x_t) - f(x^*)] \\
&\overset{③}{\leq} E[\|x_t - x^*\|^2] + 2\eta(3\eta L_s - 1)E[f(x_t) - f(x^*)] + 6\eta^2 L_s E[\langle \pi_{\mathcal{I}}(\nabla f(x^*)), x_t - x^*\rangle] \\
&+ \frac{3\eta^2}{b_t}\sigma_0^2 + 3\eta^2 E[\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(u_t)\|^2],
\end{aligned}
$$

where ① holds because u_t is independent of all other random variables, such as x_t, and $E[u_t] = 0$; ② holds because $E[\langle x_t - x^*, \pi_{\mathcal{I}}(g_t)\rangle] \geq E[f(x_t) - f(x^*)]$, which is derived from restricted strong convexity, ③ holds by Lemma 4 in [47] that

$$
E[\|\pi_{\mathcal{I}}(g_t)\|^2] \leq 6L_s E[f(x_t) - f(x^*)] + 6L_s E[\langle \pi_{\mathcal{I}}(\nabla f(x^*)), x_t - x^*\rangle] + \frac{3}{b_t}\sigma_0^2 + 3\,\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2\,,
$$

and σ_0^2 is defined as in Assumption 2.

By the restricted ρ_s-strong convexity and setting $\eta \leq \frac{1}{3L_s}$ yields

$$
\begin{aligned}
E[\|y_t - x^*\|^2] \leq (1 + \rho_s\eta(3\eta L_s - 1))E[\|x_t - x^*\|^2] + 2\eta(6\eta L_s - 1)E[\langle \nabla_{\mathcal{I}} f(x^*), x_t - x^*\rangle] + \frac{3\eta^2}{b_t}\sigma_0^2 \\
+ 3\eta^2 E[\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2] + \eta^2 E[\|\pi_{\mathcal{I}}(u_t)\|^2].
\end{aligned}
$$

Here the operator $\pi_I(x)$, as defined in Section 3, zeros out the elements of x not indexed in $\mathcal{I}$. Because the size of support $\mathcal{I}$ is $3k$ and $u_t \sim N(0, \sigma^2\mathbf{I})$, we have $E[\|\pi_I(u_t)\|^2] \leq 3k\sigma^2$. Then if $\eta = \frac{1}{6L_s}$, we get

$$
E[\|y_t - x^*\|^2] \leq (1 - \frac{1}{12\kappa_s})E[\|x_t - x^*\|^2] + \frac{1}{12L_s^2}E[\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2] + \frac{1}{b_t}\frac{1}{12L_s^2}\sigma_0^2 + \frac{k}{12L_s^2}\sigma^2.
$$

The following result [26] shows that the HT operator is nearly non-expensive when k is much larger than optimal sparsity $k*$.

$$
\|\mathcal{H}_k(x) - x^*\|_2^2 \leq (1 + \beta)\|x - x^*\|_2^2, \tag{4}
$$

for $k > k^*$ and for any parameter $x \in \mathbb{R}^d$, where $\beta = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}}$ and $k^* = \|x^*\|_0$.

Then we can obtain

$$
\begin{aligned}
E[\|x_{t+1} - x^*\|^2] &\leq (1 + \frac{2\sqrt{k^*}}{\sqrt{k - k^*}})E[\|y_t - x^*\|^2] \\
&= \theta_1 E\,\|x_t - x^*\|^2 + \frac{1+\beta}{12L_s^2}E[\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2] + \frac{1+\beta}{12L_s^2 b_t}\sigma_0^2 + \frac{k(1+\beta)}{12L_s^2}\sigma^2,
\end{aligned}
$$

where $\theta_1 = (1 + \frac{2\sqrt{k^*}}{\sqrt{k-k^*}})(1 - \frac{1}{12\kappa_s})$ and $\beta = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}}$. If we further require $\theta_1 = (1 + \frac{2\sqrt{k^*}}{\sqrt{k-k^*}})(1 - \frac{1}{12\kappa_s}) < 1$, and $\frac{1}{b_t}\frac{1+\beta}{12L_s^2}\sigma_0^2 \leq \frac{k(1+\beta)}{12L_s^2}\sigma^2$, i.e. $k \geq 4k^*(12\kappa_s - 1)^2 + k^*$, and $\sigma_0^2 \leq kb_t\sigma^2$, we get

$$
\begin{aligned}
E[\|x_T - x^*\|^2] &\leq \theta_1^T\,\|x_0 - x^*\|^2 + \frac{1-\theta_1^{T-1}}{1-\theta_1}\frac{1+\beta}{12L_s^2}\left\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\right\|^2 + \frac{1-\theta_1^{T-1}}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\sigma^2. \\
&\leq \theta_1^T\,\|x_0 - x^*\|^2 + \frac{1}{1-\theta_1}\frac{1+\beta}{12L_s^2}\left\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\right\|^2 + \frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\sigma^2.
\end{aligned}
$$

□

Theorem 4.2 shows that the DP-SGD-HT converges to x^* with an estimation error bias in a linear convergence rate and the convergence factor is specified by θ_1. This result is consistent with the non-DP SGD-HT [33] and HSGD-HT [47], which both achieve a linear convergence rate. Precisely, the estimation error is upper bounded by the sum of three terms in Eq.(3). The first term approaches 0 when the number of iterations t goes to infinity. The second term is a statistical bias term due to the sparsity constraint on the solution x^*. If x^* is sufficiently close to the unconstrained minimizer of f when k is chosen to be large, then $\|\nabla f(x^*)\|$ becomes close to 0. The final term is a bias term generated by the Gaussian mechanism's perturbation noise, which ensures differential privacy. When this noise specified by σ^2 approaches 0, the third term vanishes. The second and third terms together form an estimation error floor that does not vanish with increasing iterations. Compared with the original SGD-HT algorithm [33], the upper bound Eq.(3) incurs an additional term determined by σ^2. However, our analysis no longer requires that the condition number $\kappa_s \leq \frac{4}{3}$ in [33], which is difficult to satisfy.

4.3. The Utility Bound of the DP-SGD-HT

Designing algorithms that satisfy the DP requires a tradeoff between the utility of the algorithm and the level of data privacy preservation. As a result, it is critical to investigate how privacy preservation and algorithm convergence are related, or how the inclusion of a randomized process to the update rule of the algorithm affects the performance of the optimization algorithm. To evaluate the utility of our DP-SGD-HT algorithm, we are interested in knowing how closely the iterate of the algorithm x_T approaches an optimal solution x^*, i.e., the quantity $E[\|x_T - x^*\|^2]$. The smaller this quantity is, the better. In our utility analysis - Theorem 4.3, we show that the utility of Algorithm 1 is reserved because the convergence rate is bounded by the sum of two terms, the first of which is related to the bias due to the original algorithm's sparsity requirement, and the second to the DP parameters.

Theorem 4.3. *Under the same setting of Theorem 4.2, if we let $T = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$, the output of Algorithm 1, x_T, satisfies*

$$E[\|x_T - x^*\|^2] \leq \frac{1}{1-\theta_1}\frac{(1+\beta)}{12L_s^2}\left\|\pi_{\tilde{I}}(\nabla f(x^*))\right\|^2 + O(\frac{\log(1/\delta)}{n^2\epsilon^2}\log(\frac{n^2\epsilon^2}{\log(1/\delta)})). \tag{5}$$

Here the expectation is taken over all the randomness of the algorithm, including both the subsampling for computing stochastic gradients and the random noise added for ensuring differential privacy.

Proof. From Theorem 4.2, we have

$$E[\|x_T - x^*\|^2] \leq \underbrace{\theta_1^T\|x_0 - x^*\|^2}_{①} + \underbrace{\frac{1}{1-\theta_1}\frac{1+\beta}{12L_s^2}\left\|\pi_{\tilde{I}}(\nabla f(x^*))\right\|^2}_{②} + \underbrace{\frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\sigma^2}_{③}. \tag{6}$$

The third term ③ is determined by the noise level σ^2, which is fixed for a given noise level σ^2. The first term ① is related to the number of iterations T and can decay to zero for large T. However, when the first term ① is less than the third term ③, having more iterations may not improve the bound further. (Note that the second term is due to the sparsity of the solution which is not an amenable algorithm parameter.) Therefore, let term ① ≤ term ③ and we can obtain the optimal choice of T.

Setting $\theta_1^T\|x_0 - x^*\|^2 \leq \frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\sigma^2 = \frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\frac{40\alpha l^2}{n^2\epsilon}$ yields

$$\begin{aligned}
T &= \log_{\theta_1}\left(\frac{1}{\|x_0 - x^*\|^2}\frac{1}{1-\theta_1}\frac{k(1+\beta)}{6L_s^2}\frac{40\alpha l^2}{n^2\epsilon}\right) \\
&= \log\left(\|x_0 - x^*\|^2(1-\theta_1)\frac{6L_s^2}{k(1+\beta)}\frac{n^2\epsilon}{40\alpha l^2}\right)/\log(1/\theta_1) \\
&= \log\left(\|x_0 - x^*\|^2(1-\theta_1)\frac{6L_s^2}{k(1+\beta)}\frac{n^2\epsilon}{40(1+\frac{2\log(1/\delta)}{\epsilon})l^2}\right)/\log(1/\theta_1) \\
&= O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)})).
\end{aligned}$$

where the second equation is by the change of base formula of logarithms: $\log_a(b) = \log(1/b)/\log(1/a)$ for $0 < a < 1$ and $0 < \theta_1 < 1$, the third equation is due to $\alpha = 1 + \frac{2\log(1/\delta)}{\epsilon}$. Once we remove all constants, we get the final result.

Hence, when $T = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$ we get the upper bound of $E[\|x_T - x^*\|^2]$ in Eq. 5. □

Remark 4.4. *Theorem 4.3 implies that the DP-SGD-HT approximates a sparse optimal solution with an upper bound of* $O(\left\|\pi_{\tilde{I}}(\nabla f(x^*))\right\|^2 + \frac{\log(1/\delta)}{n^2\epsilon^2})$*. The term* $O(\|\pi_{\tilde{I}}(\nabla f(x^*))\|^2)$ *specifies the sparsity-induced statistical error, which approaches* 0 *if* x^* *is sufficiently close to an unconstrained minimizer of* $f(x)$*, so it represents the sparsity-induced bias to the solution of the unconstrained optimization problem. The second term* $O(\frac{\log(1/\delta)}{n^2\epsilon^2})$ *is induced by the Gaussian mechanism and will be large with small* ϵ *and* δ*, which is corresponding to the high privacy guarantee situation, and hence plays the dominating role in high privacy regime.*

Based on the convergence analysis, we can further analyze the computational complexity of the DP-SGD-HT, which specifies an upper bound on the total number of computations of pair $(f_z(x), \nabla f_z(x))$ that Algorithm 1 needs to calculate during the training process in Corollary 4.4.1.

Corollary 4.4.1 (Computational Complexity)**.** *Under the same conditions of Theorem 4.3, its computational complexity is* $T \times b = O(b\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$.

Note that early analysis of the DP-GD-HT shows that the computational complexity of the non-stochastic version is in the order of $O(n\log(n))$ [42]. Our stochastic version with a computational complexity of $O(b\log(n))$ is better because the size of mini-batch b is generally much smaller than the training sample size n.

5. The DP-SCSG-HT

Although the proposed DP-SGD-HT method significantly reduces the computational cost compared to full gradient methods, the randomness in sampling the mini-batches introduces additional variance into gradient estimation. We now perturb the stochastically controlled stochastic gradients in DP-SCSG-HT using variance reduction techniques, which improves and accelerates convergence and utility over DP-SGD-HT. In particular, the variance of stochastic gradients can be well controlled by full or large-batch gradients calculated at each snapshot in a variance reduction technique. Because full gradients may waste computation, as discussed in [17], we calculate a batch gradient to correct the mini-batch stochastic gradients once in several iterations. The number of iterations in the inner loop is determined by a geometric distribution because we use the stochastically controlled stochastic gradient method.

Definition 5 (Geometric Distribution)**.** *A random variable N follows a geometric distribution Geom*(γ)*, denoted as* $N \sim$ *Geom*(γ)*, if N is a non-negative integer and the probability distribution is* $P(N = k) = (1-\gamma)\gamma^k, \ \ \forall k = 0, 1, \cdots$ *Then, we have the expectation of N,* $E[N] = \frac{\gamma}{1-\gamma}$.

The DP-SCSG-HT has two loops, as shown in Algorithm 2: the outer loop (Lines 2 - 16) and the inner loop (Lines 9 - 14). To approximate the full gradient, a batch gradient is computed at each outer iteration (Line 5), so the batch size B is set to be large. Stochastic gradients are calculated on mini-batches with a much smaller size b in an inner loop. In contrast to the DP-SGD-HT algorithm, the number of iterations in the inner loop $N^{(j)}$ requires to be determined, which we suggest two options: in option I, $N^{(j)}$ is randomly drawn from a geometric distribution, similar to the methods in [24, 14]; in option II, a deterministic constant $\frac{B}{b}$ is used and $\frac{B}{b}$ is the expectation of the geometric distribution Geom $(B/(B+b))$. In practice, both options are applicable, and as observed in [24, 14], option II can be more stable, because setting $N^{(j)}$ to a constant eliminates the variance of $N^{(j)}$ introduced in option I. However, with option I, the property of geometric distribution makes the theoretical analysis of Algorithm 2 more concise. Thus, we perform the theoretical analyses of the DP-SCSG-HT method based on both of the options, which provide a more general setting for both theoretical analyses and practical applications.

5.1. Differential Privacy Guarantee of the DP-SCSG-HT

DP analysis can be difficult for DP-SCSG-HT, because the number of inner iterations N is a random variable that must be bound using the geometric distribution property. We first show that the proposed DP-SCSG-HT algorithm satisfies the (α, ρ)-RDP, which is then converted into the (ϵ, δ)-DP as summarized in Theorem 5.1. Different from the

Algorithm 2 DP-SCSG-HT

```
**Input:** The maximal number of outer loops $\mathcal{J}$, initial state $\tilde{x}^1$, stepsize $\eta$, batch sizes $B$, and $b$, $\sigma_1$, and $\sigma_2$
**for** $j = 1, 2, ..\mathcal{J}$ **do**
    Randomly pick $I^{(j)} \subset \{1, ..., n\}$, where $|I^{(j)}| = B$
    $u_1^{(j)} \sim N(0, \sigma_1^2 \mathbf{I})$
    $\tilde{\mu}^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)}) + u_1^{(j)}$
    $x_0^{(j)} = \tilde{x}^{(j)}$
    **option I:** Generate $N^{(j)} \sim$ Geom $(B/(B+b))$
    **option II:** $N^{(j)} = \frac{B}{b}$
    **for** $t = 1, 2, \ldots, N^{(j)}$ **do**
        Randomly pick $I_t^{(j)} \subset \{1, ..., n\}$, where $|I_t^{(j)}| = b$
        $u_{t,2}^{(j)} \sim N(0, \sigma_2^2 \mathbf{I})$
        $v_t^{(j)} = \nabla f_{I_t^{(j)}}(x_t^{(j)}) - \nabla f_{I_t^{(j)}}(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)} + u_{t,2}^{(j)}$
        $x_t^{(j)} = \mathcal{H}_k(x_{t-1}^{(j)} - \eta v_t^{(j)})$
    **end for**
    set $\tilde{x}^{j+1} = x_{N^{(j)}}^{(j)}$
**end for**
```

analysis of the DP-SGD-HT, every updating iteration based on the stochastic variance reduced gradient deals with two different subsampling: $I^{(j)}$ at a snapshot and $I_t^{(j)}$ at each iteration of the inner loop. A proof sketch is provided below and more details are given in the Appendix.

Theorem 5.1. *Let the maximal number of epochs be $\mathcal{J}$, and $\frac{\sigma_1^2}{160} = \frac{\sigma_2^2}{40} = \sigma^2$ where $\sigma^2 = \frac{2CBl^2\alpha\mathcal{J}}{bn^2\epsilon}$ for a constant $C > 0$, and $\alpha = 1 + \frac{2\log(2/\delta)}{\epsilon}$. Algorithm 2 satisfies the (ϵ, δ)–DP if $\alpha \leq \log(\frac{bn^3\epsilon}{Bbn^2\epsilon + 20CB^4\alpha\mathcal{J}})$, $\frac{20\alpha CBb\mathcal{J}}{n^2\epsilon} \geq 1.5$ and $1 - (1 - \frac{\delta}{2})^{\frac{1}{\mathcal{J}}} \geq e^{-(C-1-\ln(C))}$.*

Proof Sketch: Let S be a set of n training examples. We consider the following two queries:

$$\tilde{q}_{t,1}^{(j)}(S) = \nabla f_{I^{(j)}}(\tilde{x}^{(j)}),$$

$$\tilde{q}_{t,2}^{(j)}(S) = \nabla f_{I_t^{(j)}}(x_{t-1}^{(j)}) - \nabla f_{I_t^{(j)}}(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)},$$

given $\tilde{\mu}^{(j)}$.

Part I. For $\tilde{q}_{t,1}^{(j)}(S)$, we consider the following query function: $q_{t,1}^{(j)}(S) = \frac{1}{B}\sum_{z=1}^{n} \nabla f_z(\tilde{x}^{(j)})$. By Lemma 3.2, for query function $q_1^{(j)}(S)$, the Gaussian mechanism $\mathcal{M}_1 = q_{t,1}^{(j)}(S) + u_1^{(j)}$, where $u_1^{(j)} \sim N(0, \sigma_1^2 \mathbf{I})$ is $(\alpha, \frac{\alpha\Delta_2^2(q_1^{(j)})}{2\sigma_1^2})$-RDP, and is more precisely $(\alpha, \frac{4\alpha l^2}{B^2\sigma_1^2})$-RDP.

Then, let us examine the subsampling query $\tilde{q}_1^{(j)}(S)$. The mechanism $\tilde{\mathcal{M}_1}^{(j)} = \tilde{q}_1^{(j)}(S) + u_1^{(j)}$ is $(\alpha, \frac{20\alpha l^2}{n^2\sigma_1^2})$-RDP, if $\alpha \leq \log(\frac{n}{B(1+\sigma_1^2 B^2/4l^2)})$ and $\frac{B^2\sigma_1^2}{4l^2} \geq 1.5$.

Part II. For $\tilde{q}_{t,2}^{(j)}(S)$, we first examine the following query function: $q_{t,2}^{(j)}(S) = \frac{1}{b}\sum_{z=1}^{n} \nabla f_z(x_{t-1}^{(j)}) - \frac{1}{b}\sum_{z=1}^{n} \nabla f_z(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)}$, conditioning on $\tilde{\mu}^{(j)}$. By Lemma 3.2, for the query function $q_{t,2}^{(j)}(S)$, the Gaussian mechanism $\mathcal{M}_2 = q_{t,2}^{(j)}(S) + u_{t,2}^{(j)}$, where $u_{t,2}^{(j)} \sim N(0, \sigma_2^2 \mathbf{I})$ is $(\alpha, \frac{\alpha\Delta_2^2(q_{t,2}^{(j)})}{2\sigma_2^2})$-RDP.

Then, we examine the following query with the subsample $I_t^{(j)}$, $\tilde{q}_{t,2}^{(j)}(S) = \nabla f_{I_t^{(j)}}(x_{t-1}^{(j)}) - \nabla f_{I_t^{(j)}}(\tilde{x}^{(j)}) + \tilde{\mu}^{(j)}$ conditioning on $\tilde{\mu}^{(j)}$. The mechanism $\tilde{\mathcal{M}_2} = \tilde{q}_{t,2}^{(j)}(S) + u_{t,2}^{(j)}$ is $(\alpha, \frac{80\alpha l^2}{n^2\sigma_2^2})$-RDP, if $\alpha \leq \log(\frac{16nl^2}{16l^2 b + \sigma_2^2 b^3})$ and $\frac{b^2\sigma_1^2}{16l^2} \geq 1.5$.

Combining the analyses of **Part I** and **Part II**, and setting $\frac{\sigma_1^2}{160} = \frac{\sigma_2^2}{40} = \sigma^2$, yield that $(\tilde{\mathcal{M}_1}, \tilde{\mathcal{M}_2})$ satisfies $(\alpha, \frac{l^2\alpha}{n^2\sigma^2})$-RDP, by the composition rule in Lemma 3.3.

Because $N^{(j)} \sim Geom(B/(B+b))$ is a random variable, we need to bound $N^{(j)}$ in order to apply the composition rule. Hence, we consider the event $\mathbb{E} = \{N^{(j)} \leq \frac{CB}{b}$ for $1 \leq j \leq \mathcal{J}\}$ with the probability of $\mathbb{E}$ as $\mathbb{P}(\mathbb{E})$. We prove that there exists a constant C satisfying $e^{-(C-1-\ln(C))} \leq 1-(1-\frac{\delta}{2})^{\frac{1}{\mathcal{J}}}$, such that the number of $N^{(j)}$ is upper bounded by $\frac{CB}{b}$ with at least the probability $(1-\frac{\delta}{2})^{\frac{1}{\mathcal{J}}}$. Hence, $\mathbb{P}(\mathbb{E}) = \Pi_{j=1}^{\mathcal{J}} P(\mathbb{E}_j) \geq 1-\frac{\delta}{2}$ where $\mathbb{E}_j$ is the event of $N^{(j)} \leq \frac{CB}{b}$ for $\forall j$. Conditioning on event $\mathbb{E}$, we can show that Algorithm 2 satisfies the $(\frac{CB\alpha l^2\mathcal{J}}{bn^2\sigma^2}+\frac{\log(2/\delta)}{\alpha-1}, \delta/2)$-DP, which is the $(\epsilon, \delta/2)$-DP if $\alpha = 1+\frac{2\log(2/\delta)}{\epsilon}$ and $\sigma^2 = \frac{2CB\alpha l^2\mathcal{J}}{bn^2\epsilon}$. By Definition 1, for adjacent datasets S, S' and any output O, we obtain $\mathbb{P}[\mathcal{M}(S)\in O|\mathbb{E}] \leq e^{\epsilon}\cdot\mathbb{P}[\mathcal{M}(S')\in O|\mathbb{E}]+\delta/2$. Therefore, we further obtain

$$\begin{aligned}
&\mathbb{P}[\mathcal{M}(S)\in O]\\
&=\mathbb{P}[\mathcal{M}(S)\in O|\mathbb{E}]\cdot\mathbb{P}(\mathbb{E})+\mathbb{P}[\mathcal{M}(S)\in O|\mathbb{E}^c]\cdot\mathbb{P}(\mathbb{E}^c)\\
&\leq (e^{\epsilon}\cdot\mathbb{P}\left[\mathcal{M}(S')\in O\middle|\,\mathbb{E}\right]+\delta/2)\mathbb{P}(\mathbb{E})+\delta/2\\
&\leq e^{\epsilon}\cdot\mathbb{P}\left[\mathcal{M}(S')\in O\middle|\,\mathbb{E}\right]\cdot\mathbb{P}(\mathbb{E})+\delta\\
&\leq e^{\epsilon}\cdot\mathbb{P}\left[\mathcal{M}(S')\in O\right]+\delta,
\end{aligned}$$

where $\mathbb{E}^c$ is the complementary event of $\mathbb{E}$. Therefore, Algorithm 2 satisfies the (ϵ, δ)-DP. □

For option II, analysis becomes easier because the number of iterations in an epoch ($N^{(j)}$) is fixed and the composition rule for RDP can be directly applied. We can easily show that the DP-SCSG-HT with option II also satisfies the DP with a constant $C = 1$, $\alpha = 1+\frac{2\log(1/\delta)}{\epsilon}$.

Remark 5.2. *The variance of the injected Gaussian noise is required to be $\sigma^2 = \frac{40\alpha l^2 T}{n^2\epsilon}$ for the DP-SGD-HT where T is the total number of iterations. Compared with the DP-SGD-HT, the variances of Gaussian noises in the DP-SCSG-HT σ_1^2 and σ_2^2 satisfy $\frac{\sigma_1^2}{160} = \frac{\sigma_2^2}{40} = \sigma^2$, and the value of $\sigma^2 = \frac{2CBl^2\alpha\mathcal{J}}{bn^2\epsilon}$ can be much smaller. If $C = 1$ for option II, the number of total inner iterations for the DP-SCSG-HT is $\frac{B\mathcal{J}}{b}$, which is usually smaller than T in practice for large-scale problems. Therefore, practically, DP-SCSG-HT achieves better estimation, due to the lower bias derived from the lower perturbation noise, as analyzed in Theorem 4.2 and empirically observed in the experiments section.*

5.2. Convergence Guarantee of the DP-SCSG-HT

We examine how adding Gaussian noises in Algorithm 2 to preserve data privacy can alter the convergence of the algorithm. We develop an upper bound on the distance between the estimator x_t and the optimal x^*.

Theorem 5.3. *Suppose that $f(x)$ satisfies Assumptions 1 and 3. Define $\tilde{\mathcal{I}} = supp(x^*) \cup supp(\mathcal{H}_{2k}(\nabla f(x^*)))$. Let $k^* = \|x^*\|_0$, the restricted condition number of $f(x)$, $\kappa_s = \frac{L_s}{\rho_s} \geq 1$, and $\beta = \frac{2\sqrt{k^*}}{\sqrt{k-k^*}} \leq min\{\frac{b}{B}, \frac{1}{64\kappa_s^2-1}\}$. If the variance of stochastic gradients $\mathbb{I}(B<n)\sigma_0^2 \leq kB\sigma^2$, then we can get,*

$$E[\|\tilde{x}^{(j+1)}-x^*\|^2] \leq \theta_2^{j+1}\|\tilde{x}^{(0)}-x^*\|^2+\frac{1}{128(1-\theta_2)\gamma L_s^2\kappa_s^2}\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2+\frac{7k}{1024(1-\theta_2)\gamma L_s^2\kappa_s^2}\sigma^2, \tag{7}$$

where $\theta_2 = 1-\frac{1}{\frac{64\kappa_s^2}{1+\beta}\frac{b}{B}+\frac{13}{8}} < 1$, $\gamma = \frac{\frac{b}{B}-\beta}{1+\beta}+\frac{14\kappa_s-1}{512\kappa_s^3} > 0$ and $\mathbb{I}(\cdot)$ is an indicator function.

Proof sketch: We first give some preparations and then show the line of main proof.

1) Preparations. In our analysis, we introduce an error term $e^{(j)} = \nabla f_{I^{(j)}}(\tilde{x}^{(j)})-\nabla f(\tilde{x}^{(j)})$, which plays an important role in the flow of the derivation, and is one of the major differences from the analysis of the existing SVRG-HT [27]. Because $v_t^{(j)} = \nabla f_{I_t^{(j)}}(x_t^{(j)})-\nabla f_{I_t^{(j)}}(\tilde{x}^{(j)})+\tilde{\mu}^{(j)}+u_{t,2}^{(j)}$ is the updating direction at the t^{th} iteration of the j^{th} epoch in Algorithm 2, $e^{(j)}$ is the bias of the updating direction $v_t^{(j)}$, where $E_{I_t^{(j)}}[v_t^{(j)}] = \nabla f(x_t^{(j)})+e^{(j)}$ and $E_{I_t^{(j)}}$ is the expectation over stochastic sampling $I_t^{(j)}$. We show that the variance of the error term $e^{(j)}$ can be bounded as

$$E[\|\pi_{\mathcal{I}}(e^{(j)})\|^2] \leq 2L_s^2\frac{\mathbb{I}(B<n)}{B}E[\|\tilde{x}^{(j)}-x^*\|^2]+2\frac{\mathbb{I}(B<n)}{B}\sigma_0^2, \tag{8}$$

which will diminish to zero with an increasing batch size B. The above bound gives extra flexibility to adaptively adjust the batch size B based on the variance of Gaussian perturbed noise.

Before diving into the detailed proof, we also need to analyze the term for the variance of stochastic gradient direction - $E_{I_t^{(j)}}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2]$ on $\pi_{\mathcal{I}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$, which is a projection operator to support $\mathcal{I}$. Then we have:

$$\begin{aligned} E_{I_t^{(j)}}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] \leq\ & 4L_s(f(x^*) - f(x_t^{(j)})) + 4L_s(f(x_0^{(j)}) - f(x^*)) + 4L_s(\langle \pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^* \rangle) \\ & + 2\|\pi_{\mathcal{I}}(\nabla f(x^*))\|^2 + 2\|\pi_{\mathcal{I}}(\nabla f(x_t^{(j)}))\|^2 + 2L_s^2\|x_0^{(j)} - x^*\|^2 + 2\|\pi_{\mathcal{I}}(e^{(j)})\|^2 + E[\|\pi_{\mathcal{I}}(u_t^{(j)})\|^2], \end{aligned} \tag{9}$$

where $e^{(j)}$ is the bias of $v_t^{(j)}$. The Eq.(9) indicates that the variance of stochastic gradient direction can diminish to zero, when the model estimator $x^{(j)}$ is approaching to optimal x^* and x^* is close to solution of the unconstrained problem (1), as long as both $\|\pi_{\mathcal{I}}(e^{(j)})\|^2$ and $E[\|\pi_{\mathcal{I}}(u_t^{(j)})\|^2]$ are small.

2) Proof. With above preparations, we are ready to give the logic line of proof for main theorem. In order to analyze the DP-SCSG-HT algorithm, we develop the following result,

$$E_{I_t^{(j)}}[\|\tilde{x}_{t+1}^{(j)} - x^*\|^2] = E_{I_t^{(j)}}[\|x_t^{(j)} - x^*\|^2] + \eta^2 E_{I_t^{(j)}}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] - 2\eta\langle \pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^* \rangle - 2\eta\langle \pi_{\mathcal{I}}(e^{(j)}), x_t^{(j)} - x^* \rangle$$

where $\tilde{x}_{t+1}^{(j)} = x_t^{(j)} - \eta\pi_{\mathcal{I}}(v_t^{(j)})$ is an intermediate state of the estimator to bridge the analysis between the gradient-based updating step and the hard thresholding step. Then the hard thresholding operation $x_{t+1}^{(j)} = \mathcal{H}_k(\tilde{x}_{t+1}^{(j)})$ immediately follows and we can get $x_{t+1}^{(j)} = \mathcal{H}_k(x_t^{(j)} - \eta v_t^{(j)})$ due to $\mathcal{I} = supp(x^*) \cup supp(x_t^{(j)}) \cup supp(x_{t+1}^{(j)})$.

Next, we establish connections between the intermediate state $\tilde{x}_{t+1}^{(j)}$ and the sparse estimator $x_{t+1}^{(j)}$. By Eq.(4), we get

$$\begin{aligned} E^{(j)}[\|x_{t+1}^{(j)} - x^*\|^2] &\leq (1+\beta)E^{(j)}[\|\tilde{x}_{t+1}^{(j)} - x^*\|^2] \\ &\leq (1+\beta)E^{(j)}[\|x_t^{(j)} - x^*\|^2] + (1+\beta)\eta^2 E^{(j)}[\|\pi_{\mathcal{I}}(v_t^{(j)})\|^2] - 2(1+\beta)\eta E^{(j)}[\langle \pi_{\mathcal{I}}(\nabla f(x_t^{(j)})), x_t^{(j)} - x^* \rangle]. \end{aligned} \tag{10}$$

Until now, all the analyses are still based on iterations in one epoch. We need to use an important property of the geometric distribution that we have used to set the number of inner iterations $N^{(j)}$ to turn previous iteration-based analysis into the epoch-based analysis. Let $N \sim Geom(\gamma)$, for any sequence $\{D_N\}$, we have $E[D_N - D_{N+1}] = (\frac{1}{\gamma} - 1)(D_0 - E[D_N])$. Taking the expectation on both sides of Eq. (10) over $N^{(j)}$, and replacing $x_0^{(j)}$ with $\tilde{x}^{(j)}$ and $x_{N^{(j)}}^{(j)}$ with $\tilde{x}^{(j+1)}$ yields the most important intermediate result:

$$\begin{aligned} & 2(1+\beta)\eta E[\langle \pi_{\mathcal{I}}(\nabla f(\tilde{x}^{(j+1)})), \tilde{x}^{(j+1)} - x^* \rangle] \\ & \leq (\beta - \frac{b}{B})E[\|\tilde{x}^{(j+1)} - x^*\|^2] + \frac{b}{B}E[\|\tilde{x}^{(j)} - x^*\|^2] + (1+\beta)\eta^2 E[\|\pi_{\mathcal{I}}(v_{N^{(j)}}^{(j)})\|^2]. \end{aligned} \tag{11}$$

After obtaining the above results, we put Eq. (9) for $E[\|\pi_{\mathcal{I}}(v_{N^{(j)}}^{(j)})\|^2]$ into Eq. (11) and further using ρ_s-restricted strongly convex and L_s-restricted strongly smooth, we obtain the desired result. □

Remark 5.4. *Due to the requirement on β that $\frac{B}{b} \leq \frac{1}{\beta} = \frac{\sqrt{k-k^*}}{2\sqrt{k^*}} = \Theta(\sqrt{k})$, $\frac{B}{b}$ is independent of the sample size n. Hence, $\frac{B}{b}$ can be treated as a constant independent of ϵ and δ, and be omitted in the asymptotic utility bound in the next sections.*

The implication of the main theorem is that the variance of stochastic gradients σ_0^2 can be well-controlled by the batch size B, and the requirement for the upper bound of the stochastic variance σ_0^2 will be relaxed with the increase of B and be removed when $B = n$. Therefore, unlike the DP-SGD-HT, there is no need to bound σ_0^2 in Algorithm 2 with $B = n$. Nevertheless, a careful setup for B could save the computational cost. Even though setting $B = n$ could fully remove σ_0^2, it needs to be carefully designed to achieve the best of the two worlds, which means that the effect of σ_0^2 is minimized to the distance bound between estimator x_T and optimal x^*, and the batch size B is also minimized to achieve such goal to avoid the waste of computations.

Similar to the analysis of the DP-SGD-HT, the two bias terms in the parameter estimation of DP-SCSG-HT are the second term and third term of Eq. (7): $O(\frac{\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2}{\kappa_s^2} + \frac{\sigma^2}{\kappa_s^2})$. Compared to the bias of the DP-SGD-HT

$O(\left\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\right\|^2 + \sigma^2)$, the bias of the DP-SCSG-HT shrinks by a factor of κ_s^2. The value of κ_s^2 is greater than 1 and can be very large for ill-conditioned optimization problems. As discussed in Remark 5.2, the variance of Gaussian noise is also smaller in the DP-SCSG-HT than in the DP-SGD-HT in practice. Hence, the DP-SCSG-HT tends to have smaller bias in terms of parameter estimation.

5.3. The Utility Bound of the DP-SCSG-HT

Using the upper bound on the distance between $\tilde{x}^{(\mathcal{J})}$ and the optimal x^* in Theorem 5.3, and the determined Gaussian variance σ^2 in Theorem 5.1, we can obtain the utility bound as follows.

Theorem 5.5. *Under the same setting of Theorem 5.3, and $B = \max\{1, \sqrt{\frac{2b\epsilon\sigma_0^2}{3k\alpha C\mathcal{J}l^2}}\}\cdot n$, if we choose $\mathcal{J} = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$, we get*

$$E[\|\tilde{x}^{(\mathcal{J})} - x^*\|^2] \leq \frac{8\eta^2}{(1-\theta_2)\gamma}\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2 + O(\frac{\log(1/\delta)}{n^2\epsilon^2}\log(\frac{n^2\epsilon^2}{\log(1/\delta)})). \tag{12}$$

Proof. If we require that $\mathbb{I}(B < n)\sigma_0^2 \leq kB\sigma^2$ and $\sigma^2 = \frac{2CBl^2\alpha\mathcal{J}}{bn^2\epsilon}$, we have $B = \min\{1, \sqrt{\frac{2b\epsilon\sigma_0^2}{3k\alpha C\mathcal{J}l^2}}\} * n$.

$$\begin{aligned} E[\|\tilde{x}^{(\mathcal{J})} - x^*\|^2] &\leq \theta_2^{\mathcal{J}} E[\|\tilde{x}^{(0)} - x^*\|^2] + \frac{8\eta^2}{(1-\theta_2)\gamma}E[\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2] + \frac{7k\eta^2\sigma^2}{(1-\theta_2)\gamma} \\ &= \theta_2^{\mathcal{J}}\|\tilde{x}^{(0)} - x^*\|^2 + \frac{8\eta^2}{(1-\theta_2)\gamma}E[\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2] + \frac{1}{1-\theta_2}\frac{7k\eta^2}{\gamma}\frac{2CBl^2\alpha\mathcal{J}}{bn^2\epsilon}. \end{aligned}$$

If we let $\theta_2^{\mathcal{J}}\|\tilde{x}^{(0)} - x^*\|^2 \leq \frac{1}{1-\theta_2}\frac{7k\eta^2}{\gamma}\frac{2CBl^2\alpha}{bn^2\epsilon}$ and $\frac{B}{b} = \Theta(\sqrt{k})$, we get

$$\mathcal{J} = \log_{\theta_2}(\frac{1}{(1-\theta_2)\|\tilde{x}^{(0)} - x^*\|^2}\frac{7k\eta^2}{\gamma}\frac{2CBl^2\alpha}{bn^2\epsilon}) = O(\log(\frac{n^2\epsilon^2}{\log(1/\delta)})).$$

Finally, we get

$$E[\|\tilde{x}^{(\mathcal{J})} - x^*\|^2] \leq \frac{8\eta^2}{(1-\theta_2)\gamma}\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2 + O(\frac{\log(1/\delta)}{n^2\epsilon^2}\log(\frac{n^2\epsilon^2}{\log(1/\delta)})).$$

□

Considering that $\|\pi_{\tilde{\mathcal{I}}}(\nabla f(x^*))\|^2$ can be close to zero, when x^* is close to its unconstrained optional for $f(x)$, Theorem 5.5 implies that the utility bound is determined by its dominant term in the order of $O(\frac{\log(1/\delta)}{n^2\epsilon^2})$, which achieves the same utility guarantee with DP-GD-HT. Furthermore, the next corollary shows better computationally complexity of our proposed practical stochastic variance reduced algorithm.

Corollary 5.5.1 (Computational Complexity)**.** *Under the same conditions of Theorem 5.5, its computational complexity is $O(\min\{1,\psi\}\cdot n\log(\frac{n^2\epsilon^2}{\log(1/\delta)}))$, where $\psi = \sqrt{\frac{2b\epsilon\sigma_0^2}{3k\alpha C\mathcal{J}l^2}}$.*

To obtain a given $(\epsilon,\delta)-$DP, the computational complexity of DP-SCSG-HT depends on $O(\min\{1,\psi\}\cdot n\log(n))$. Because $\sigma_0^2 \leq l^2$, $\alpha > 1$ and $C > 1$, ψ can be much smaller than 1, if sparsity k and epoch size $\mathcal{J}$ are large, batch size b is small (it is especially true for high dimensional data). Therefore, similar to DP-SGD-HT, DP-SCSG-HT can be much more computationally efficient than DP-GD-HT, which means fewer number of epochs are used in Algorithm 2 to achieve the same $(\epsilon,\delta)-$DP.

In summary, our proposed algorithm provides a general framework, which covers the existing state-of-the-art non-DP hard thresholding method: SVRG-HT [27] (when $B = n$, $b = 1$, $\sigma^2 = 0$ and option II is selected), which corresponding to privacy-preserving version can be called as DP-SVRG-HT. Even though we only use option I to theoretically analyze Algorithm 2 for clarity, our DP guarantee can be directly applied to option II and so is to DP-SVRG-HT. Following the line of proof above, the convergence analysis for DP-SVRG-HT can be done, and then utility bound can be built in the same way in section 5.3.

6. Empirical Evaluations

We implement the proposed stochastic privacy-preserving algorithms DP-SGD-HT and DP-SCSG-HT using pyTorch, and compare them with the state-of-the-art DP methods - DP-GD-HT [42] - to evaluate the performance of both accuracy and computational efficiency. The comparison algorithm DP-GD-HT has been implemented based on the design in [42], and applied to our study datasets. Moreover, two widely used non-DP HT methods, SVRG-HT [27] and SCSG-HT [29] are also used to produce the non-DP baseline performance. Because SVRG-HT can be treated as a special case of SCSG-HT (when the batch size $B = n$, and also discussed in Table 1), and SCSG-HT(with $B \neq n$) outperforms SVRG-HT as discussed in [29] , we include SCSG-HT in our experiments. It should be noted that non-DP methods are expected to produce higher accuracy because they are not constrained by the DP. Our DP versions, on the other hand, must find a balance between model accuracy and the risk of data leakage. In recent years, federated learning has become a popular area to utilize edge computing devices to perform large scale decentralized machine learning. However, when an analytic model is built during such a learning process, it may raise separate concerns about data leakage. To further investigate the performance of our stochastic privacy-preserving algorithms, we implement one of our proposed algorithms in the FL setting.

6.1. Experimental Setup

Two benchmark datasets, E2006-tfidf and RCV1, are downloaded from the LibSVM website[1], and used for evaluation. We also conduct experiments on the Chest X-ray [22] medical dataset.

- **The E2006-tfidf dataset** has 3,308 observations, each described by 150,360 features. The dataset is obtained from Noah Smith and is uesd to predict the volatility of stock returns based on the mandated financial text report. Data has been collected from thousands of publicly traded U.S. companies.
- **The RCV1 dataset** contains 20,242 observations and 47,236 features. The RCV1 dataset is a benchmark dataset on text categorization. It is a collection of newswire articles produced by Reuters in 1996-1997, and categorized with respect to three controlled vocabularies: industries, topics and regions.
- **The Chest X-ray dataset** has 5232 records and is used to detect pneumonia from each record based on 784 image features. In [22], a collection of 5232 chest X-ray images was gathered from children, including 3,883 characterized as depicting pneumonia (2,538 bacterial and 1,345 viral) and 1,349 normal, from a total of 5,856 patients to train a predictive classifier. The model can then be tested with 234 normal images and 390 pneumonia images(242 bacterial and 148 viral) from 624 patients.

In the experiments, the variance of the injected random noise is determined based on the values given by the algorithms' theoretical results. Other parameters, such as the batch size, the stepsize and the number of epochs, are determined by five-fold cross validation. Particularly, the stepsize η for each algorithm is searched from $\{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and the number of epochs is searched from $\{10, 20, 50, 80, 100\}$. All the algorithms are initialized with $x^{(0)} = 0$. Following the convention in the stochastic optimization and sparse learning literature, we use the number of epochs (or data passes) to measure the computational complexity. This enables the complexity study independent of an actual implementation of the algorithm. All experiments are done on PC with i7-6700 CPU, 4 cores, 8GB RAM.

6.2. Linear Regression

We first conduct experiments on the linear regression problem

$$\min_x \{f(x) = \frac{1}{n}\sum_{i=1}^{n} \|y_i - z_i^T x\|^2\} \text{ subject to } \|x\|_0 \leq k,$$

to check the performance of the proposed DP-SGD-HT and DP-SCSG-HT algorithms. The dataset we use is the E2006-tfidf dataset. In the experiments, we set the sparsity parameter $k = 200$, $\delta = 10^{-5}$ and $\epsilon \in [2, 10]$. Table 3

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

Table 3: Comparison of different algorithms given different privacy budgets ϵ in terms of MSE on the validation data of five-fold cross validation and its corresponding standard deviation on the dataset E2006-tfidf. We use $\delta = 10^{-5}$ in the experiment. The non-DP Baseline is obtained by SCSG-HT [29], which reflects the state-of-the-art of non-DP IHT algorithms. Each column represents one group of experiment for a fixed privacy guarantee (ϵ, δ)-DP. The number of epochs (Epoch) is used to measure computational complexity. The results show that DP-SCSG-HT achieves the lowest MSE among DP algorithms and is closer to the non-DP baseline.

Methods	Epoch	Differential private budget ϵ				
		$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$
Non-DP Baseline SCSG-HT [29]	10	0.1483 ± 0.013	0.1483 ± 0.013	0.1483 ± 0.013	0.1483 ± 0.013	0.1483 ± 0.013
DP-GD-HT[42]	100	0.1588 ± 0.025	0.1566 ± 0.015	0.1560 ± 0.009	0.1543 ± 0.01	0.1528 ± 0.012
DP-SGD-HT	20	0.1540 ± 0.005	0.1505 ± 0.009	0.1499 ± 0.013	0.1488 ± 0.011	0.1490 ± 0.013
DP-SCSG-HT	10	**0.1516** ± **0.007**	**0.1494** ± **0.014**	**0.1488** ± **0.007**	**0.1487** ± **0.006**	**0.1486** ± **0.012**

Table 4: Comparison of different algorithms given different privacy budgets ϵ in terms of the validation loss (13) on validation data of five-fold cross validation and its corresponding standard deviation on dataset RCV1. Note that $\delta = 10^{-5}$ in the experiment. The results show that DP-SCSG-HT achieves the lowest validation loss with the smallest number of epochs, among DP methods.

Methods	Epoch	Differential private budget ϵ				
		$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$
Non-DP Baseline SCSG-HT [29]	10	0.1603 ± 0.003	0.1603 ± 0.003	0.1603 ± 0.003	0.1603 ± 0.003	0.1603 ± 0.003
DP-GD-HT[42]	100	0.3811 ± 0.011	0.3469 ± 0.01	0.3139 ± 0.004	0.3063 ± 0.009	0.3001 ± 0.004
DP-SGD-HT	20	0.4100 ± 0.027	0.2914 ± 0.012	0.2594 ± 0.008	0.2615 ± 0.011	0.2491 ± 0.008
DP-SCSG-HT	10	**0.2243** ± **0.012**	**0.1662** ± **0.022**	**0.1642** ± **0.007**	**0.1619** ± **0.002**	**0.1615** ± **0.005**

compares the mean squared errors (MSE) of the different methods on validation data under different choices of privacy budget ϵ. In a five-fold cross validation process, the MSE values are averaged across the five validation sets together with standard deviation. Precisely, MSE on a single validation set is defined as follows: $\frac{1}{n_{val}}||Z_{val}^T\tilde{x} - y_{val}||^2$, where $\{Z_{val}, y_{val}\}$ are the validation data, n_{val} is the validation sample size and $\tilde{x}$ is the estimator learned from the training data. The results in Table 3 show that under the same guarantee of (ϵ, δ)-DP, the proposed methods: DP-SGD-HT and DP-SCSG-HT achieve lower MSE using a smaller number of epochs than the DP-GD-HT. Therefore, the utility and computational complexity of our stochastic methods are better than that of the non-stochastic DP-GD-HT. DP algorithms take the balance between privacy-preserving degree and optimization accuracy, but our algorithms exhibit better accuracy even under the DP requirement.

6.3. Logistic Regression

Then, we apply all methods to the logistic regression problem as follows

$$\min_x\{f(x) = \frac{1}{n}\sum_{i=1}^{n}(\log(1 + exp(y_i z_i^T x)) + \frac{\lambda}{2}||x||^2)\} \text{ subject to } ||x||_0 \leq k,$$

where $z_i \in \mathbb{R}^d$ and y_i is the corresponding label. The dataset we use are the RCV1 datasetand Chest X-ray. For experiment with RCV1, the regularizer $\lambda = 10^{-5}$ and the sparsity parameter $k = 1000$. For experiment with Chest X-ray, the regularizer $\lambda = 10^{-5}$ and the sparsity parameter $k = 200$. We use five-fold cross-validation to calculate the value of loss function:

$$\frac{1}{n_{val}}\sum_{i=1}^{n_{val}}(\log(1 + exp(y_i z_i^T x)) \tag{13}$$

Table 5: Comparisons of different algorithms for various privacy budgets ϵ in terms of the validation loss (13) on validation data of five-fold cross validation and its corresponding standard deviation on dataset Chest X-ray. Note that $\delta = 10^{-5}$ in the experiment. The results show that DP-SCSG-HT achieves the lowest validation loss with the smallest number of epochs, among DP methods, except $\epsilon = 2$.

Methods	Epoch	Differential private budget ϵ				
		$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$	$\epsilon = 10$
Non-DP Baseline SCSG-HT [29]	10	0.6917 ± 0.007	0.6917 ± 0.007	0.6917 ± 0.007	0.6917 ± 0.007	0.6917 ± 0.007
DP-GD-HT[42]	100	**0.6930 ± 0.003**	0.6929 ± 0.001	0.6928 ± 0.004	0.6928 ± 0.002	0.6927 ± 0.002
DP-SGD-HT	20	0.6932 ± 0.007	**0.6927 ± 0.002**	0.6926 ± 0.003	0.6924 ± 0.001	0.6922 ± 0.004
DP-SCSG-HT	10	0.6940 ± 0.002	0.6928 ± 0.004	**0.6919 ± 0.008**	**0.6918 ± 0.001**	**0.6918 ± 0.001**

over validation data and our proposed algorithm DP-SCSG-HT could achieve the lowest loss value among all privacy-preserving algorithms on the RCV1 data in Table 4.

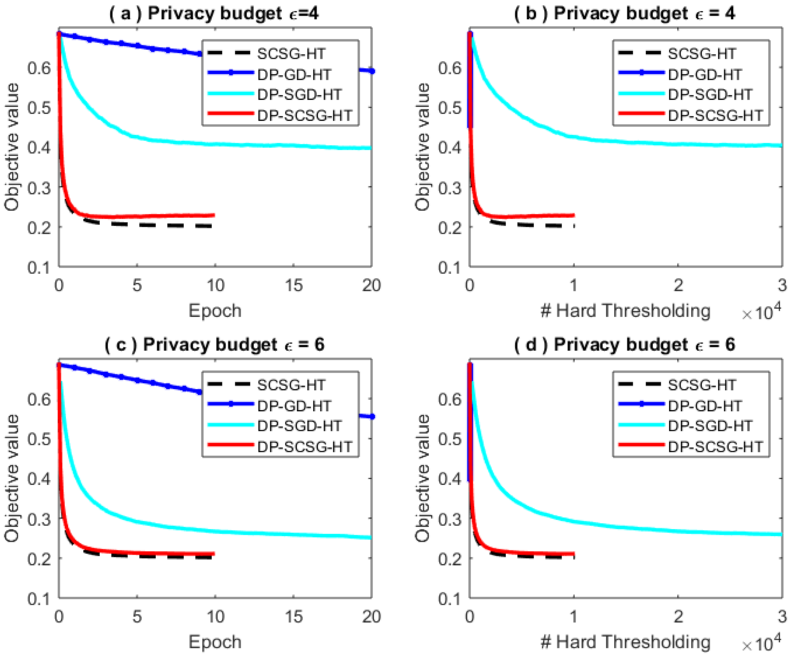

Figure 1: Experimental results for logistic regression with sparsity constraint on the RCV1 data. Figures (a-b) show results for the $(4, 10^{-5})$-DP and (c-d) for the $(6, 10^{-5})$-DP. (a) and (c) show the objective value $f(x)$ on the full dataset versus the number of epoch. (b), (d) the objective value $f(x)$ on the full dataset versus the number of HT operations.

Separate from the five-fold cross validation, we run all algorithms on the full dataset so to compare the computational efficiency of the different algorithms. We demonstrate the advantage of the stochastic algorithms by plotting the objective function value $f(x)$ versus the number of epochs and the number of hard thresholding operations of different algorithms at the privacy budget $\epsilon \in \{4, 6\}$ on RCV1 in Figure 1 and Chest X-ray in Figure 2 . Our algorithms outperform the deterministic DP-GD-HT in terms of the needed epochs by a large margin, which is consistent with our theoretical results. While DP-SCSG-HT and DP-SGD-HT achieve the best results within 20 epochs, DP-GD-HT needs much more epochs. Therefore, the proposed algorithms could drop objective function value more rapidly, while guaranteeing the (ϵ, δ)−DP.

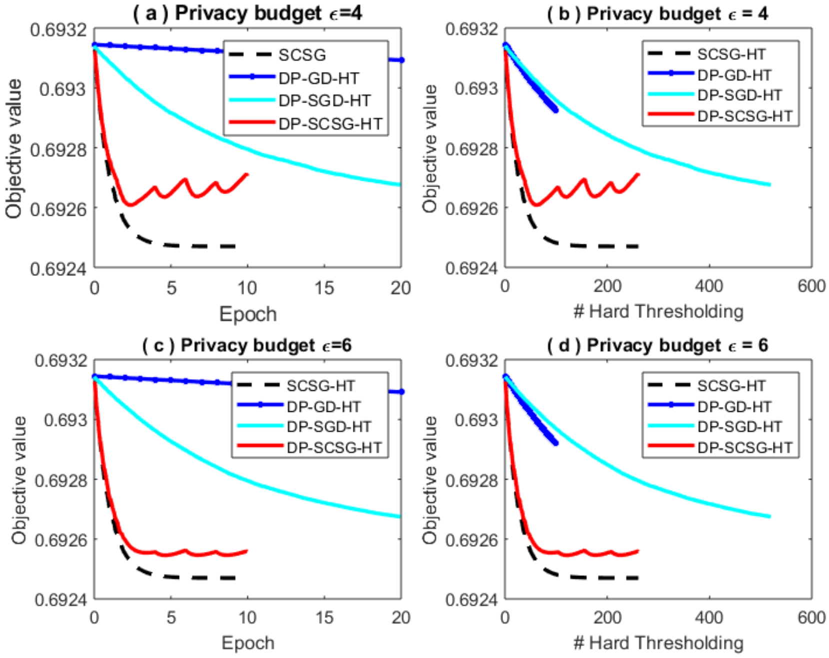

Figure 2: Experimental results for logistic regression with sparsity constraint on the Chest X-ray data. Figures (a-b) show results for the (4, 10^{-5})-DP and (c-d) for the (6, 10^{-5})-DP. (a) and (c) show the objective value $f(x)$ on the full dataset versus the number of epoch. (b), (d) the objective value $f(x)$ on the full dataset versus the number of HT operations.

6.4. Federated Learning Algorithms with DP

Federated learning (FL) is a privacy-preserving learning framework for large scale machine learning on edge computing devices, and solves the data-decentralized distributed optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) = \sum_{i=1}^{N} p_i f_i(x), \tag{14}$$

where $f_i(x) = E_{z\sim\mathcal{D}_i}[f_z^i(x)]$ is the loss function of the i^{th} client (or device) with weight $p_i \in [0, 1)$, $\sum_{i=1}^N p_i = 1$, $f_z^i(x)$ (for $z \sim \mathcal{D}_i$) is an individual loss associated with the z^{th} sample in i^{th} client, $\mathcal{D}_i$ is the distribution of data located locally on the i^{th} client, and N is the total number of clients. The weights p_i can be necessary to balance the different clients if the clients participating the FL vary significantly in terms of computing capability and carry different amounts of local data. Nowadays, FL is getting more attention and it is an important question to ask if FL can also benefit from our DP methods [13, 3]. We thus evaluate the performance of our DP method in the FL setting. We implement a federated learning version of DP-SGD-HT. (We leave the FL implementation of DP-SCSG-HT to future research because it requires more careful design for the different FL schemes of SCSG which is outside of this paper's scope.)

In this set of experiments, we compare three FL algorithms: the FL version of standard SGD (FedSGD-HT), the FL version of our DP-SGD-HT (DP-FedSGD-HT), and a specific FL implementation of DP-GD-HT (DP-FedGD-HT). The detailed steps of these FL algorithms are included in Appendix C. Our algorithm DP-FedSGD-HT randomly samples a mini-batch from the local dataset at each client and computes stochastic gradient directions using this local mini-batch. Each client performs K consecutive SGD iterations with perturbed stochastic gradients for local update before communicating with other clients. For DP-FedGD-HT, each client calculates local full gradients using all local data and performs K consecutive local GD iterations. Again, we could remove the Gaussian noise perturbation in DP-FedSGD-HT, which gives the non-DP method FedSGD-HT.

The MNIST dataset [9] is used in this set of experiments because MNIST data have been decentralized with the sort-and-partition procedure (SP) [25, 37]. Each device contains data on two digits. We use a convolutional neural

Table 6: The architecture of CNN for the MNIST dataset.

layer	layer setting
F.relu(self.conv1(x))	self.conv1 = nn.Conv2d(1, 6, 5)
F.max_pool2d(x, 2, 2)	
F.relu(self.conv2(x))	self.conv2 = nn.Conv2d(6, 16, 5)
x.view(-1, 16*4)	
F.relu(self.fc1(x))	self.fc1 = nn.Linear(16*4*4, 120)
x= F.relu(self.fc2(x))	self.fc2 = nn.Linear(120, 84)
x = self.fc3(x)	self.fc3 = nn.Linear(84, 10)
F.log_softmax(x, dim=1)	

network (CNN) with two convolutional layers and three fully connected layers (see Table 6). We set the number of clients $N = 10$, the number of local updates $K = 10$ for each client, the sparsity parameter $\tau = 10,000$. The Gaussian noises are generated according to moments accountant technique using TensorFlow-privacy [1] for local privacy budget $\epsilon = 4, 6$.

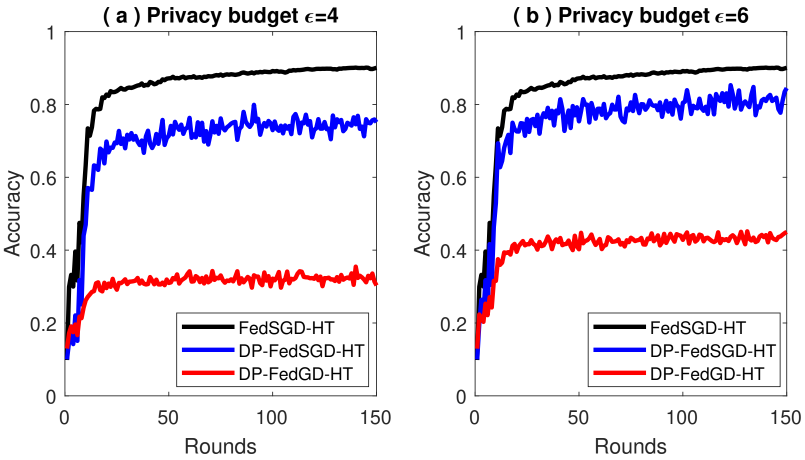

Figure 3: Experimental results for federated learning of CNN models with sparsity constraint on the MNIST data. Figure (a) shows results for the (4, 10^{-5})-DP and (b) for the (6, 10^{-5})-DP. Both (a) and (b) show the accuracy on the MNIST dataset versus the number of rounds.

From Figure 3, we clearly observe that the stochastic DP algorithm DP-FedSGD-HT outperforms DP-FedGD-HT, which is consistent with our experimental results for linear regression and logistic regression in the early sections. The worse performance of DP-FedGD-HT might also be partially due to the nonconvexity of CNN model training, for which deterministic GD algorithms could easily get stuck at bad local minimal. The DP version of the FedSGD-HT is slightly worse than the non-DP version as expected, and with larger privacy budget ($\epsilon = 6$), the difference between the two methods decreases.

Remark 6.1. *The guarantee of (ϵ, δ)-DP adds an extra constraint to the optimization algorithm; Therefore, DP algorithms preserve privacy at the cost of losing prediction accuracy (utility). Empirically, we did observe that our algorithms slightly sacrifice prediction accuracy, and the performance gap with the non-DP baseline is reduced with larger privacy budget ϵ by using a smaller level of injected Gaussian noise. Hence, the algorithms play balance between privacy preserving and utility. The observations are consistent with prior discussions in [42] .*

7. Conclusions

In this paper, we propose two iterative hard thresholding algorithms for sparse learning that preserve privacy: DP-SGD-HT and DP-SCSG-HT. To balance between DP and the algorithmic utility, the proposed algorithms play trade-off between the magnitude of perturbation noise (the privacy budget) and the convergence rate. The greater the perturbation noise, the greater the privacy preservation but the lower the algorithm utility (i.e., larger bound on the convergence speed). Under certain biases introduced by sparsity and perturbation noise, we establish a linear convergence rate for both algorithms. The best known utility bound is achieved by our algorithms. Meanwhile they reduce the computational complexity of the GD-based algorithm significantly. We emphasize that, although the DP-SGD-HT convergence proof requires the variance of stochastic gradients to be bounded, this requirement is removed in the DP-SCSG-HT convergence proof. Experiments on real-world financial and medical datasets demonstrate the superiority of the proposed algorithms against the state-of-the-art baseline algorithms.

Acknowledgments

This work was partially supported by the US National Science Foundation (NSF) under grants: CCF-1514357 and IIS-1718738, and National Institutes of Health (NIH) under grants 5K02DA043063-03 and R01-DA051922-01 to Jinbo Bi. The work of J. Ding and M. Pan was supported in part by the U.S. NSF under grants CNS-1646607, CNS-1801925, and CNS-2029569.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[3] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific reports*, 12(1):1–10, 2022.

[4] S. Bahmani, B. Raj, and P. T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(Mar): 807–841, 2013.

[5] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[6] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3): 265–274, 2009.

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

[8] A. E. C. Cloud. Amazon web services. *Retrieved November*, 9(2011):2011, 2011.

[9] L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[12] C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[13] A. El Ouadrhiri and A. Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022.

[14] M. Elibol, L. Lei, and M. I. Jordan. Variance reduction with sparse gradients. *arXiv preprint arXiv:2001.09623*, 2020.

[15] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[16] J. Guo, X. Ding, T. Wang, and W. Jia. Combinatorial resources auction in decentralized edge-thing systems using blockchain and differential privacy. *Information Sciences*, 2022.

[17] R. B. Harikandeh, M. O. Ahmed, A. Virani, M. Schmidt, J. Konečnỳ, and S. Sallinen. Stopwasting my gradients: Practical svrg. In *Advances in Neural Information Processing Systems*, pages 2251–2259, 2015.

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

[19] X. Jiang, C. Niu, C. Ying, F. Wu, and Y. Luo. Pricing gan-based data generators under rényi differential privacy. *Information Sciences*, 602: 57–74, 2022.

[20] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[21] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[22] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

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

[24] L. Lei and M. Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156, 2017.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[26] X. Li, R. Arora, H. Liu, J. Haupt, and T. Zhao. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*, 2016.

[27] X. Li, T. Zhao, R. Arora, H. Liu, and J. Haupt. Stochastic variance reduced optimization for nonconvex sparse learning. In *International Conference on Machine Learning*, pages 917–925, 2016.

[28] X. Li, H. Li, H. Zhu, and M. Huang. The optimal upper bound of the number of queries for laplace mechanism under differential privacy. *Information Sciences*, 503:219–237, 2019.

[29] G. Liang, Q. Tong, C. J. Zhu, and J. Bi. An effective hard thresholding method based on stochastic variance reduction for nonconvex sparse learning. In *AAAI*, pages 1585–1592, 2020.

[30] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[31] I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[32] J. Near. Differential privacy at scale: Uber and berkeley collaboration. In *Enigma 2018 (Enigma 2018)*, 2018.

[33] N. Nguyen, D. Needell, and T. Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63(11):6869–6895, 2017.

[34] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[35] A. Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

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

[37] Q. Tong, G. Liang, T. Zhu, and J. Bi. Federated nonconvex sparse learning. *arXiv preprint arXiv:2101.00052*, 2020.

[38] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.

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

[40] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[41] H. Wang and H. Wang. Correlated tuple data release via differential privacy. *Information Sciences*, 560:347–369, 2021.

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

[43] L. Wang and Q. Gu. A knowledge transfer framework for differentially private sparse learning. *arXiv preprint arXiv:1909.06322*, 2019.

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

[45] Y.-X. Wang, B. Balle, and S. Kasiviswanathan. Subsampled r\'enyi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*, 2018.

[46] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1307–1322, 2017.

[47] P. Zhou, X. Yuan, and J. Feng. Efficient stochastic gradient hard thresholding. In *Advances in Neural Information Processing Systems*, pages 1988–1997, 2018.

[48] J. Zhu, X. Fang, Z. Guo, M. H. Niu, F. Cao, S. Yue, and Q. Y. Liu. Ibm cloud computing powering a smarter planet. In *IEEE International Conference on Cloud Computing*, pages 621–625. Springer, 2009.