Collect at Once, Use Effectively: Making Non-interactive Locally Private Learning Possible

Kai Zheng[*1] Wenlong Mou[*1] Liwei Wang[1]

Abstract

Non-interactive Local Differential Privacy (LDP) requires data analysts to collect data from users through noisy channel at once. In this paper, we extend the frontiers of Non-interactive LDP learning and estimation from several aspects. For learning with smooth generalized linear losses, we propose an approximate stochastic gradient oracle estimated from non-interactive LDP channel using Chebyshev expansion, which is combined with inexact gradient methods to obtain an efficient algorithm with quasi-polynomial sample complexity bound. For the high-dimensional world, we discover that under ℓ_2-norm assumption on data points, high-dimensional sparse linear regression and mean estimation can be achieved with logarithmic dependence on dimension, using random projection and approximate recovery. We also extend our methods to Kernel Ridge Regression. Our work is the first one that makes learning and estimation possible for a broad range of learning tasks under non-interactive LDP model.

1. Introduction

Data privacy has become an increasingly important issue in the age of data science. Differential Privacy (DP), proposed in 2006 by Dwork et al.,(Dwork et al., 2006), provide a solid foundation and rigorous standard for private data analysis. Since then, there has been extensive literature studying the fundamental trade-offs between differential privacy and accuracy for query answering (Hardt & Rothblum, 2010; Hardt et al., 2012; Thaler et al., 2012; Wang et al., 2016), machine learning (Chaudhuri & Monteleoni, 2008; Chaudhuri et al., 2011; Rubinstein et al., 2012; Wang et al., 2015), and statistical inference (Lei, 2011; Smith, 2011). For more details on DP results, please refer to the excellent monograph written by Dwork and Roth (Dwork & Roth, 2014). Intuitively, a DP algorithm uses randomized response to defend against adversary, so that change of one of data points could not be detected.

Despite the prevailing success of this notion in academia, its applicability in data science practice could be limited. For example, if data analysts just promise to follow the differential privacy constraints, user will not feel their privacy are preserved. The promise could not be validated; the mechanisms are complicated; and even worse: users do not trust the data collector at all. Unfortunately, most of differential privacy algorithms are based on adding noise calibrated to stability of loss function, which essentially requires access to original data.

Borrowing ideas from classical wisdom on collecting sensitive survey data (Warner, 1965), Local Differential Privacy (LDP) (Kasiviswanathan et al., 2008; Duchi et al., 2013b) was proposed as a stronger notion of privacy to resolve this problem. LDP requires each of data points to be passed through a noisy channel during collection. This channel will ensure one can hardly tell anything about the user based on what he have sent. The practical advantage of LDP is obvious: users will be comfortable sending their sensitive information through noisy channels, which are transparent and reliable; additionally, users can choose their own privacy parameters, making it possible to associate with economic value. Therefore, this line of research has attracted lots of attention (Duchi et al., 2013a;b; Kairouz et al., 2014; Bassily & Smith, 2015; Kairouz et al., 2016).

Despite the analogy in definition, the way in which LDP achieves accurate results are fundamentally different from classical DP. Essentially, the information collected from each user is almost completely noisy, from which one needs to obtain accurate results. The only way to do that is to make the independently distributed noise cancel out with each other in some sense. With sand being washed away by waves, golds begin to appear.

[*]Equal contribution [1]Key Laboratory of Machine Perception, MOE, School of EECS, Peking University, Beijing, China. Correspondence to: Kai Zheng <zhengk92@pku.edu.cn>, Wenlong Mou <mouwenlong@pku.edu.cn>, Liwei Wang <wanglw@cis.pku.edu.cn>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

Two local privacy notions have been discussed in existing literature: the interactive model allows the algorithm to collect data sequentially, and decide what to ask based on information from previously asked users. The non-interactive model, on the contrary, requires all data to be collected at once, with no interactive queries allowed. Apparently the non-interactive model is strictly stronger, and prohibition on interactive queries rules out most of SGD-type approaches, making the problem significantly harder. However, non-interactive LDP is more useful in real-world applications, as opportunities of interactive queries may not be available in most settings.

In existing literature, learning and inference under interactive and non-interactive LDP therefore are exhibiting different appearances. In the interactive world, LDP is promised with connection to Statistical Query (SQ) model (Kearns, 1998), from its very beginning (Kasiviswanathan et al., 2011). SQ algorithms for a wide range of convex ERM problems were proposed by (Feldman et al., 2017), implying good risk bounds for LDP. (Duchi et al., 2013b) established matching upper and lower bounds for convex risk minimization problems. On the other hand, very few has been done in the non-interactive setting. Existing works primarily focus on basic estimation problems such as means and discrete densities (Duchi et al., 2013a; 2016; Bassily & Smith, 2015), or some function calculations (Kairouz et al., 2015b). Most of important modern learning and inference tasks, including estimation in linear models and convex ERM, are still poorly understood in non-interactive local DP settings.

For the high-dimensional world, where $d \gg n$ while some low-complexity constraints are imposed, we may hope the error induced by privacy constraints to be logarithmically dependent upon d. In classical differential privacy literature, this has been be addressed using different techniques, guarantee error bounds logarithmically dependent on dimension (Talwar et al., 2015; Smith & Thakurta, 2013). However, lower bounds have been shown in local privacy model even for high-dimensional 1-sparse mean estimation, ruling out any good guarantees (Duchi et al., 2016). The lower bound result illustrates fundamental difficulties of local differential privacy. But if we still want to do high-dimensional learning under local privacy, are there additional assumptions that helps?

Therefore, the starting point of this work lies on making learning possible under the non-interactive LDP setting, which is the hardest yet the most useful. We initiate the first attempt towards a broad range of learning tasks beyond simple distribution estimation. In particular, we investigate two important classes of problems under non-interactive LDP: (1) High-dimensional sparse linear regression and mean estimation; (2) Generalized linear models. Our focus is to design corresponding mechanisms and study their convergence rates with respect to the number and dimension of data. One can also consider optimal mechanisms in terms of privacy parameters like (Geng & Viswanath, 2014), which is of independent interests.

Our Contributions: In this paper, we propose several efficient algorithms for learning and estimation problems under non-interactive LDP model, with good theoretical guarantees. In the following we summarize our contributions.

(1) High Dimensional Estimation: One of exciting findings in this paper is about local privacy for high-dimensional data. Roughly speaking, convergence rate with logarithmic dependence on the dimension can be attained under LDP, if we assume data points are ℓ_2 bounded. This is in sharp contrast with information-theoretic lower bounds for 1-sparse mean estimation for ℓ_∞ bounded data (Duchi et al., 2016). Valid algorithms are presented for both sparse mean estimation and sparse linear regression, respectively. Intuitively, non-interactivity doesn't bring about additional difficulties, since the loss functions are quadratic forms. However, if we directly add noise to each of data points and send it to the server, the aggregated noise will lead to linear dependence on the dimension. Thus we adopt the random projection technique, and send the noisy version of projected data to the server. Based on the aggregated information, we can approximately recover the optimal solution via linear inverse problem.

(2) Learning Smooth Generalized Linear Models: Generalized linear problems which has additional smooth properties (we call the loss with respect to it as smooth generalized linear loss (SGLL), see rigorous definition in section 2) include many common loss functions, such as logistic loss, square loss, etc. Optimizing such losses are intuitively much more difficult in non-interactive LDP model, as the loss can be an arbitrary function $\boldsymbol{w}^T\boldsymbol{x}$. This even makes it difficult for us to obtain an unbiased estimator for objective function, or its gradient. As a result, when we aggregate the loss of noisy data together, it is even hard to ensure it converge to the population loss. Approximation theory techniques are introduced to tackle this problem. In particular, we use polynomials of $\boldsymbol{w}^T\boldsymbol{x}$ to approximate nonlinear coefficients of gradients. Chebyshev bases, instead of Taylor series, are used to get faster convergence within an arbitrary domain. Then we are able to build inexact stochastic gradient oracles to arbitrarily specified accuracy. SIGM algorithm in (Dvurechensky & Gasnikov, 2016) is exploited to find the minimizer with inexact gradients.

Other Related Work: Local privacy dates back to (Warner, 1965), who uses random responses to protect privacy in surveys. In recent LDP literature, both (Duchi et al., 2013a) and (Kairouz et al., 2016) studied density estimation methods and their theoretical behaviors in

LDP model. Rather than statistical setting in above two work, (Bassily & Smith, 2015) considered how to produce frequent items and corresponding frequencies of a dataset in local model. Besides, (Kairouz et al., 2014) investigated optimality of LDP mechanisms based on information theoretical measures for statistical discrimination.

Approximation techniques are commonly used in DP literature. (Thaler et al., 2012) employed polynomials for marginal queries. (Wang et al., 2016) leveraged trigonometric polynomials to answer smooth queries. (Zhang et al., 2012) also used polynomial approximations and get basic convergence results in standard DP model. Besides, the random projection and recovery has also been used in DP learning (Kasiviswanathan & Jin, 2016) and local DP histogram estimation (Bassily & Smith, 2015).

In standard DP model, both high-dimensional sparse estimation and generalized linear model have been intensively studied. (Kifer et al., 2012) and (Smith & Thakurta, 2013) considered the convergence of private LASSO estimator under RSC and incoherence assumptions. (Talwar et al., 2015) considered constrained ERM of sparse linear regression, and obtained $\tilde{O}(\log d/n^{2/3})$ rate using private Frank-Wolfe. Above results assume ℓ_∞-bounded data. By stronger assumption of ℓ_2 bounded data, (Kasiviswanathan & Jin, 2016) gave a general framework for high dimensional empirical risk minimization (ERM) problem. There are several works to estimate generalized linear model under DP, with a particular emphasis on logistic regression. Objective and output perturbation are used to get low excess risks (Chaudhuri & Monteleoni, 2008; Chaudhuri et al., 2011). Both (Bassily et al., 2014) and (Zhang et al., 2017) considered concrete private algorithms to solve ERM. None of these existing results extends directly to non-interactive LDP setting.

2. Preliminaries

Some notations: $[p] = \{1, 2, \cdots, p\}$. Vectors are written in bold symbol, such as $\boldsymbol{x}, \boldsymbol{w}$. x represents univariate number, which has no relation with $\boldsymbol{x}$. For a vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T$, $\boldsymbol{x}^k$ represents the power of each element. $B_2(r) = \{\boldsymbol{x} | \left\|\boldsymbol{x}\right\|_2 \leqslant r\}$. Denote $\mathbb{S}^+$ as the semipositive matrix space, $\mathrm{Proj}_{\mathbb{S}^+}(\cdot)$ means projecting a matrix to $\mathbb{S}^+$ in terms of Frobenius norm (i.e. eliminate all negative eigenvalues). For an univariate function $f(x)$, $f^{(k)}(x)$ represents its k-th derivative, and define $\left\|f^{(k)}\right\|_T := \int_{-1}^{1} \frac{|f^{(k+1)}(x)|}{\sqrt{1-x^2}} \mathrm{d}x$. For the reason of limited space, all omitted proof can be found in the supplementary.

2.1. Local Differential Privacy

Here we adopt the LDP definition given in (Bassily & Smith, 2015).

Definition 1. *A mechanism $\mathcal{Q} : \mathcal{V} \to \mathcal{Z}$ is said to be (ϵ, δ)-local differential private or (ϵ, δ)-LDP, if for any $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}$, and any (measurable) subset $S \subset \mathcal{Z}$, there is*

$$\boldsymbol{Pr}[Q(\boldsymbol{v}) \in S] \leqslant e^{\epsilon} \boldsymbol{Pr}[Q(\boldsymbol{v}') \in S] + \delta$$

Just the same with basic results in DP (Dwork & Roth, 2014), there are corresponding basic results for LDP:

Lemma 1 (Gaussian Mechanism)**.** *If $\mathcal{V} = \{\boldsymbol{v} \in \mathbb{R}^d | \left\|\boldsymbol{v}\right\|_2 \leqslant 1\}$, then $\mathcal{Q}(\boldsymbol{v}) = \boldsymbol{v} + \boldsymbol{e}$ is (ϵ, δ)-LDP, where $\boldsymbol{e} \in \mathbb{R}^d$, and $\boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 I_d)$, $\sigma = 2\sqrt{2\ln(1.25/\delta)}/\epsilon$.*

Lemma 2 (Composition Theorem[1])**.** *Let $\mathcal{Q}_i : \mathcal{V} \to \mathcal{Z}_i$ be an (ϵ_i, δ_i)-LDP mechanism for $i \in [k]$. Then if $\mathcal{Q}_{[k]} : \mathcal{V} \to \prod_{i=1}^{k} \mathcal{Z}_i$ is defined to be $\mathcal{Q}_{[k]}(\boldsymbol{v}) = (\mathcal{Q}_1(\boldsymbol{v}), \ldots, \mathcal{Q}_k(\boldsymbol{v}))$, then $\mathcal{Q}_{[k]}$ is $(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$-LDP.*

The following simple mechanism add Gaussian noise to preserve LDP of a vector, which serves as a basic tool in LDP learning and estimation.

Algorithm 1 Basic Private Vector mechanism

```
Input: A vector $\boldsymbol{x} \in \mathbb{R}^d$, privacy parameter $\epsilon, \delta$ for LDP
Output: Private vector $\boldsymbol{z}$
Setting $\sigma = \frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}$
if $\left\|\boldsymbol{x}\right\|_2 > 1$ then
    $\boldsymbol{x} = \boldsymbol{x} / \left\|\boldsymbol{x}\right\|_2$
end if
$\boldsymbol{z} \leftarrow \boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 I_d)$
```

Theorem 1. *Algorithm 1 preserves (ϵ, δ)-LDP.*

3. High Dimensional and Non-parametric Learning via Random Projections

In this section we consider three learning problems under non-interactive LDP: Mean Estimation and Linear Regression in High-dimensions, as well as Kernel Ridge Regression. Using random projection techniques, we are able to get logarithmic dependence on d in high-dimensional settings, and also to get good guarantees for Kernel version. The first problem is considered in statistical settings, as we need to assume a sparse mean vector. The latter two problems are considered as ERM problems, which can easily be translated to population risk using uniform convergence.

3.1. High-dimensional Mean Estimation

In this section, we propose a non-interactive LDP mechanism for high-dimensional sparse mean estimation problem. By assuming ℓ_2 bounded data points, and ℓ_1 bounded

[1] Note one can also use the advanced composition mechanism (Kairouz et al., 2015a) with a refined analysis, but the main dependence over n and d will remain nearly the same.

population mean, we can get error rates with logarithmic dependence on d. Our results are in sharp contrast with the lower bound for ℓ_2-bounded general mean estimation under standard DP (Bassily et al., 2014), as well as the lower bound for ℓ_∞-bounded 1-sparse mean estimation under local DP (Duchi et al., 2016). It can be easily seen that our method extends to mean estimation problem for arbitrary low-complexity constraint set in high dimensions. We state our results in ℓ_1 setting to keep the arguments clear. Our problem adopts a statistical estimation setting as follows:

ℓ_2-bounded sparse mean estimation Suppose there is an unknown distribution $\mathcal{D}$ supported on $\mathcal{B}(0,1)$, with $\|E_{\mathcal{D}}(\boldsymbol{x})\|_1 \le \Lambda$. The ℓ_2-bounded sparse mean estimation problem requires us to produce an estimator $\hat{\boldsymbol{\theta}}$ that makes $\|\boldsymbol{\theta} - E_{\mathcal{D}}(\boldsymbol{x})\|_2$ small with high probability.

Algorithm 2 LDP ℓ_1 Constrained Mean Estimation

```
Input: $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \sim i.i.d.\mathcal{D}$
Output: Estimator $z$
  Set $p = \lceil \Lambda\epsilon\sqrt{n} \rceil$, and $m = \lceil 18 \log \frac{1}{\delta} \rceil$
  Sample $G \sim \frac{1}{\sqrt{p}}\mathcal{N}(0,1)^{p \times d}$.
  for User $i$ do
    Collect $\boldsymbol{y}_i = G\boldsymbol{x}_i + \boldsymbol{r}_i$,
    with $\boldsymbol{r}_i \sim i.i.d.\mathcal{N}(0, \frac{2\log(1.25/\delta)}{\epsilon^2} I_p)$
  end for
  for $j \in \{1, 2, \cdots, m\}$ do
    $S_j = \left\{1 + \frac{(j-1)n}{m}, 2 + \frac{(j-1)n}{m}, \cdots, \frac{jn}{m}\right\}$.
    Let $\boldsymbol{\mu}_j = \frac{1}{|S_j|}\sum_{i \in S_j} \boldsymbol{y}_i$.
  end for
  Let $\mathcal{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_m\}$.
  for $j \in \{1, 2, \cdots, m\}$ do
    Let $r_j = \min\left\{r : |\mathcal{B}_{\ell_1}(\boldsymbol{\mu}_j, r) \cap \mathcal{M}| \ge \frac{m}{2}\right\}$.
  end for
  Let $j_* = \arg\min_j r_j$, and $\boldsymbol{u} = \boldsymbol{\mu}_{j_*}$
  Solve the following convex program:
```

$$
\begin{aligned}
&\arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1 \\
&s.t. \|G\boldsymbol{z} - \boldsymbol{u}\|_1 \le \frac{100p\log(nd/\delta)}{\epsilon}\sqrt{\frac{m}{n}}
\end{aligned}
\tag{1}
$$

In Algorithm 2 we describe our data collection procedure and estimation algorithm. We are primarily using two techniques: random projection and recovery from low-complexity structures; median-of-mean estimator to boost failure probability. The privacy argument is directly implication of Theorem 1.

Intuitively, adding noise to each entry of mean vector will result in error rate's linear dependence on d. Thus we adopt the random projection technique to send a compressed version of data vector through the noisy channel. This locally private estimation procedure can be viewed as a variant of noisy compressed sensing, where ℓ_2 recovery rate is fundamentally controlled by the Gaussian Mean Width of constraint set (Vershynin, 2015). Though the distribution has bounded support, the concentration for mean estimation is dimension-dependent, while dimension-independent Markov Inequalities hold. To tackle this problem, we employ Median-of-Mean estimator to get exponential tails (Hsu & Sabato, 2016).

We first give the following bound on the error in projected space.

Lemma 3. *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \sim i.i.d.\mathcal{D}$ with $\boldsymbol{\mu} = E_{\mathcal{D}}[\boldsymbol{x}]$ and $supp(\mathcal{D}) \subseteq \mathcal{B}(0,1)$. Let G and $\{\boldsymbol{y}_i\}_{i=1}^n$ defined in the above procedure. For each of group S_j fixed, we have the following with probability $2/3$:*

$$
\left\| \frac{1}{|S_j|} \sum_{\boldsymbol{y}_i \in S_j} \boldsymbol{y}_i - G\boldsymbol{\mu} \right\|_1 \le O\left(\frac{p\log(nd)}{\epsilon\sqrt{|S_j|}} \right) \tag{2}
$$

The aggregation step in Algorithm 2 is a high-dimensional generalization of Median-of-Mean estimator used in heavy-tailed statistics. The tail properties are guaranteed in the following lemma:

Lemma 4 (Proposition 9 in (Hsu & Sabato, 2016))**.** *Suppose in metric space $\mathcal{X}$, a set of points $\mathcal{M} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_m] \sim i.i.d.\mathcal{D}$, with $Pr[d_{\mathcal{X}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}) \ge \epsilon] \le \frac{2}{3}$. Let $\hat{\boldsymbol{\theta}}$ be generated from the following procedure: $r_i = \min\left\{r : |\mathcal{B}_{\mathcal{X}}(\boldsymbol{\theta}_i, r) \cap \mathcal{M}| \ge \frac{m}{2}\right\}$,and $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}_i} r_i$. Then we have:*

$$
Pr[d_{\mathcal{X}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) \ge 3\epsilon] \le e^{-\frac{m}{18}}
$$

Since the original data are $i.i.d.$ samples from underlying distribution, small group with fixed indices should also be $i.i,d.$. Therefore $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k$ are $i.i.d.$. Combining Lemma 3 and Lemma 4 we get the following result:

Corollary 1. *The vector $\boldsymbol{u}$ constructed in Algorithm 2 satisfies the following with probability $1 - \delta$:*

$$
\|\boldsymbol{u} - G\boldsymbol{\mu}\|_1 \le O\left(\frac{p\log(nd/\delta)}{\epsilon}\sqrt{\frac{m}{n}} \right) \tag{3}
$$

Then we turn to the recovery of original mean estimator. The primary tool we are using are General M^* bound in (Vershynin, 2015).

Lemma 5 (Theorem 6.2 in (Vershynin, 2015), High Probability Version)**.** *For unknown vector $\boldsymbol{x} \in K \subseteq \mathbb{R}^d$, let $G \sim \frac{1}{\sqrt{p}}\mathcal{N}(0,1)^{p \times d}$. Noisy vector $\boldsymbol{\nu} \in \mathbb{R}^p$ with $\|\boldsymbol{\nu}\|_1 \le \sigma$. Let $\boldsymbol{y} = G\boldsymbol{x} + \boldsymbol{\nu}$. By solving the following optimization problem:*

$$
\arg\min_{\boldsymbol{x}'} \|\boldsymbol{x}'\|_K \quad s.t. \ \|G\boldsymbol{x}' - \boldsymbol{y}\|_1 \le \sigma \tag{4}
$$

where $\|\cdot\|_K$ denotes the Minkowski functional of K. Then we can get the following with probability $1-\delta$

$$\|\boldsymbol{x}-\boldsymbol{x}'\|_2 \leq O\left(\frac{w(K)+\sigma+\log\frac{1}{\delta}}{\sqrt{p}}\right)$$

where $w(K)$ denotes the Gaussian width of K.

By putting these results together we get the bound on estimation loss:

Theorem 2. *Algorithm 2 outputs $\boldsymbol{z}$ satisfying the following with probability $1-\delta$:*

$$\|\boldsymbol{z}-\boldsymbol{\mu}\|_2 \leq O\left(\log\frac{nd}{\delta}\sqrt{\log\frac{1}{\delta}}\left(\frac{\Lambda^2}{\epsilon^2 n}\right)^{\frac{1}{4}}\right)$$

3.2. Sparse Linear Regression

In this section, we consider empirical loss of sparse linear regression, i.e. $L(\boldsymbol{w};D)=\frac{1}{2n}\sum_{i=1}^{n}(\boldsymbol{x}_i^T\boldsymbol{w}-y_i)^2$, where $D=\{(\boldsymbol{x}_i,y_i)|i\in[n]\}, \|\boldsymbol{x}_i\|_2\leqslant 1, y_i\in[-1,1]$. [2].

Define $\boldsymbol{w}^*=\mathrm{argmin}_{\boldsymbol{w}\in\mathcal{C}} L(\boldsymbol{w};D)$, where $\mathcal{C}=\{\boldsymbol{w}|\,\|\boldsymbol{w}\|_1\leqslant 1\}$. We want to obtain a vector $\boldsymbol{w}^{priv}\in\mathcal{C}$ within non-interactive LDP model, such that the empirical excess risk $L(\boldsymbol{w}^{priv};D)-L(\boldsymbol{w}^*;D)$ has polynomial dependences on $\log d$ and $\frac{1}{n}$.

As in the case of high-dimensional mean estimation, directly manipulating in the original high dimensional feature space will introduce large noise, hence we use a sub-Gaussian random matrix $\Phi\in\mathbb{R}^{m\times d}$ to project original data (i.e. vectors in $\mathbb{R}^d$) into the low dimensional space (i.e. $\mathbb{R}^m$) first, then perturb each data in low dimensional space (i.e. Basic Private Vector mechanism given in Algorithm 1) which protects local privacy, and send it to the server.

Having obtained private synopsis, the server then reconstruct an unbiased estimator for objective function according to these private synopsis. We subtract a quadratic term to ensure unbiasedness and project to PSD matrices to preserve convexity. To show good approximation guarantee, we make use of RIP bounds for random projection. As the loss function is determined by inner products between $\boldsymbol{w}$ and data, it could be uniformly preserved in projected space, which guarantees the accuracy of solution estimated with local privacy. Apparently, our methods also imply bounds with general low-complexity constraint set that preserves RIP.

Our private learning mechanism is given in Algorithm 3 and any random projection matrix can be used here. The privacy argument directly follows from Private Vector Mechanism and composition.

Algorithm 3 LDP ℓ_1 Constrained Linear Regression

```
Input: Personal data $(\boldsymbol{x}, y)$, parameter $\epsilon, \delta$, projection matrix $\Phi \in \mathbb{R}^{d\times m}$
Output: Learned classifier $\boldsymbol{w}^{priv} \in \mathbb{R}^d$
for Each user $i = 1, \ldots, n$ do
    $\boldsymbol{z}_i \leftarrow$ Basic Private Vector $(\Phi^T \boldsymbol{x}_i, \epsilon/2, \delta/2)$
    $v_i \leftarrow$ Basic Private Vector $(y_i, \epsilon/2, \delta/2)$
end for
Setting $Z = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n]^T, \sigma = \frac{2\sqrt{2\ln(2.5/\delta)}}{\epsilon}$, $Q = \mathrm{Proj}_{\mathbb{S}+}(Z^T Z - n\sigma^2 I_m), \boldsymbol{v} = [v_1, \cdots, v_n]^T$
$\boldsymbol{w}^{priv} \leftarrow \mathrm{argmin}_{\boldsymbol{w}\in\mathcal{C}} \hat{L}(\boldsymbol{w}; Z, \boldsymbol{v})$, where $\hat{L}(\boldsymbol{w}; Z, \boldsymbol{v}) := \frac{1}{2n}(\Phi^T\boldsymbol{w})^T Q(\Phi^T\boldsymbol{w}) - \frac{1}{n}\boldsymbol{v}^T Z\Phi^T\boldsymbol{w}$
```

In fact, as original data is in L_2 ball, and random projection preserves norms with high probabilty, hence steps 2-4 in Algorithm 1 will be executed with very low probability.

Denote the true objective function in low dimensional space $\bar{L}(\boldsymbol{w};\bar{X},\boldsymbol{y}) := \frac{1}{2n}\left\|\bar{X}\Phi^T\boldsymbol{w}\right\|^2 - \frac{1}{n}\boldsymbol{y}^T\bar{X}\Phi^T\boldsymbol{w}$, where $\bar{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n]^T\Phi, \boldsymbol{w}\in\mathcal{C}$. Let $\hat{\boldsymbol{w}}^* := \mathrm{argmin}_{\boldsymbol{w}\in\mathcal{C}}\bar{L}(\boldsymbol{w};\bar{X},\boldsymbol{y})$. The following lemma gives the accuracy of private solution $\boldsymbol{w}^{priv}$ when reduced into low dimensional space:

Lemma 6. *Under the assumptions made in this section, given projection matrix Φ, with high probability over the randomness of private mechanism, we have*

$$\bar{L}(\boldsymbol{w}^{priv};\bar{X},\boldsymbol{y})-\bar{L}(\hat{\boldsymbol{w}}^*;\bar{X},\boldsymbol{y})\leqslant\tilde{O}\left(\sqrt{\frac{m}{n\epsilon^2}}\right) \qquad (5)$$

Now, combined with RIP bound for random projection, we can move on to prove the empirical excess risk of sparse linear regression:

Theorem 3. *Under the assumption in this section, set $m=\Theta\left(\sqrt{n\epsilon^2\log d}\right)$, then with high probability , there is*

$$L(\boldsymbol{w}^{priv})-L(\boldsymbol{w}^*)=\tilde{O}\left(\left(\frac{\log d}{n\epsilon^2}\right)^{1/4}\right)$$

Note (Talwar et al., 2015) assume data is in L_∞ ball, while both (Kasiviswanathan & Jin, 2016) and ours assume data is in L_2 ball. However, in LDP model, (Duchi et al., 2016) show it was impossible to obtain polynomial dependences over $\log d$ for ℓ_0 mean estimation problem if data is in L_∞ ball.

3.3. Infinite Dimension: Kernel Ridge Regression

Previous method mainly applies to data with finite dimensional features. However, it is common to use kernel trick in practice. This brings about new difficulties for LDP learning, as we could not add noise in the Hilbert space.

[2] Our methods suits to any radius of $\boldsymbol{x}$ and y.

In this subsection, we take kernel ridge regression as an example to show how to use Random Fourier Features (RFF) (Rahimi et al., 2007) to deal with such cases caused by shift-invariant kernels (i.e. $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x} - \boldsymbol{y})$). Note our technique also suits to similar problems.

Fix a shift-invariant kernel $k(\cdot, \cdot)$, denote the Hilbert space implicitly defined as H, and the corresponding feature map as $\Phi : \mathbb{R}^d \to H$. Let the Hilbert space corresponding to the random Fourier feature map be $\hat{H} \subset \mathbb{R}^{d_p}$, and its feature map $\hat{\Phi} : \mathbb{R}^d \to \hat{H}$, where d_p is the RFF projection dimension. Given a subset $\mathcal{X} \subset \mathbb{R}^d$ and data $D = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathcal{X}, i \in [n]\}$, for any $f \in H, g \in \hat{H}$, define loss functions in H and $\hat{H}$ as follows:

$$L_H(f) := \frac{C}{2n} \sum_i \left\| f^T \Phi(\boldsymbol{x}_i) - y_i \right\|_2^2 + \frac{1}{2} \left\| f \right\|_H^2 \qquad (6)$$

$$L_{\hat{H}}(g) := \frac{C}{2n} \sum_i \left\| g^T \hat{\Phi}(\boldsymbol{x}_i) - y_i \right\|_2^2 + \frac{1}{2} \left\| g \right\|_{\hat{H}}^2 \qquad (7)$$

where C is the regularization parameter. Denote $f^* = \operatorname{argmin}_{f \in H} L_H(f), g^* = \operatorname{argmin}_{g \in \hat{H}} L_{\hat{H}}(g), G$ as the Lipschitz constant of square loss, which depends on the bounded norm of features. Kernel ridge regression try to optimize formula (6), while after using RFF, we try to solve formula (7) in non-interactive LDP model, which can be easily tackled with similar mechanisms like sparse linear regression above. Borrow the key result in (Rubinstein et al., 2012) (restated in lemma 7 below), which used RFF to design private mechanims for SVM in DP model, it becomes easy to prove guarantees for kernel ridge regression in our setting (see Corollary 2).

Lemma 7 ((Rubinstein et al., 2012))**.** *Suppose dual variables with respect to f^*, g^* are L_1 norm bounded by some $r > 0$, and $\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}} |\Phi(\boldsymbol{x}_1)^T \Phi(\boldsymbol{x}_2) - (\hat{\Phi}(\boldsymbol{x}_1))^T \hat{\Phi}(\boldsymbol{x}_2)| \leqslant \gamma$, then there is $\sup_{\boldsymbol{x} \in \mathcal{X}} |\Phi(\boldsymbol{x})^T f^* - (\hat{\Phi}(\boldsymbol{x}))^T g^*| \leqslant r\gamma + 2\sqrt{(CG + r/2) r\gamma}$.*

Corollary 2. *Algorithm 4 satisfies (ϵ, δ)-LDP, and by setting $d_p = \tilde{O}\left(\sqrt{dn\epsilon^2}\right)$, with high probability, there is*

$$L_{\hat{H}}(\hat{\boldsymbol{w}}^{priv}) - L_H(f^*) \leqslant \tilde{O}\left(\left(\frac{d}{n\epsilon^2}\right)^{1/4}\right)$$

$$\sup_{\boldsymbol{x} \in \mathcal{X}} |\Phi(\boldsymbol{x})^T f^* - (\hat{\Phi}(\boldsymbol{x}))^T \hat{\boldsymbol{w}}^{priv}| \leqslant \tilde{O}\left(\left(\frac{d}{n\epsilon^2}\right)^{1/8}\right)$$

Algorithm 4 LDP kernel mechanism

```
Input: Personal data $(\boldsymbol{x}_i, y_i), i \in [n]$, random feature's dimension $d_p$, shift-invariant kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ with Fourier transform $f(\boldsymbol{s}) = \frac{1}{2\pi} \int e^{-j\boldsymbol{s}^T\boldsymbol{x}} k(\boldsymbol{x}) d\boldsymbol{x}$, privacy parameter $\epsilon, \delta$
Output: Private output $\hat{\boldsymbol{w}}^{priv} \in \mathbb{R}^{d_p}$
Draw i.i.d. samples $\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_{d_p} \in \mathbb{R}^d$ from $f(\boldsymbol{s})$ and $b_1, b_2, \dots, b_{d_p} \in \mathbb{R}$ from the uniform distribution on $[0, 2\pi]$
for $i = 1, \dots, n$ do
    Construct low dimensional random feature $\hat{\Phi}(\boldsymbol{x}_i) = \sqrt{\frac{1}{d_p}} \left[\cos(\boldsymbol{s}_1^T \boldsymbol{x}_i + b_1), \dots, \cos(\boldsymbol{s}_{d_p}^T \boldsymbol{x}_i + b_{d_p})\right]' \in \hat{\mathcal{C}} := \left[-\sqrt{\frac{1}{d_p}}, \sqrt{\frac{1}{d_p}}\right]^{d_p} \subset \mathbb{R}^{d_p}$
    $\boldsymbol{z}_i \leftarrow$ Basic Private Vector $(\hat{\Phi}(\boldsymbol{x}_i), \epsilon/2, \delta/2)$
    $v_i \leftarrow$ Basic Private Vector $(y_i, \epsilon/2, \delta/2)$
end for
Setting $Z = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n]^T, \sigma = \frac{2\sqrt{2\ln(2.5/\delta)}}{\epsilon}$, $Q = \text{Proj}_{\mathbb{S}+}(Z^T Z - n\sigma^2 I_{d_p}), \boldsymbol{v} = [v_1, \cdots, v_n]^T$
$\hat{\boldsymbol{w}}^{priv} \leftarrow \operatorname{argmin}_{\hat{\boldsymbol{w}}} \hat{L}(\hat{\boldsymbol{w}}; Z, \boldsymbol{v})$, where $\hat{L}(\hat{\boldsymbol{w}}; Z, \boldsymbol{v}) := \frac{1}{2n} \hat{\boldsymbol{w}}^T Q \hat{\boldsymbol{w}} - \frac{1}{n} \boldsymbol{v}^T Z \hat{\boldsymbol{w}}$
```

4. Learning Smooth Generalized Linear Model

In this section, we consider learning smooth generalized linear model in non-interactive LDP setting. Non-interactive LDP learning for this problem is essentially difficult, as it is even hard to obtain an unbiased estimator of gradient. We resolve this problem using Chebyshev polynomial expansion, which requires additional smoothness assumptions. Fortunately these assumptions are naturally satisfied by a broad range of learning tasks.

We will first define the Smooth GLM loss family with appropriate assumptions. Our definition could be shown with connection to exponential family GLM, which is commonly used in machine learning. We also illustrate our algorithm and guarantees with logistic regression.

Definition 2. *(Absolutely Smooth Functions) We say that an univariate function $h(x)$ is absolutely smooth, if for any $r > 0$, $f(x) := h(rx)$ satisfies the following properties: there exist functions $\mu_1(k; r), \mu_2(k; r)$, which are polynomial on k and $\mu_2(k; r) = O(kr)$, such that for any $k \in \mathbb{N}^+$, there is:*

(1) $f(x), f'(x), \dots, f^{(k-1)}(x)$ are absolutely continuous on $[-1, 1]$;

(2) $\left\| f^{(k)}(x) \right\|_T \leqslant \mu_1(k; r) \cdot \mu_2(k; r)^k$.

Definition 3. *(Smooth Generalized Linear Loss, SGLL) A loss function $\ell(\boldsymbol{w}; \boldsymbol{x}, y)$, is called smooth generalized linear loss, if for any given data $(\boldsymbol{x}, y)$, $\ell(\boldsymbol{w}; \boldsymbol{x}, y)$ is convex and β-smooth with respect to $\boldsymbol{w}$, and there exist absolutely smooth functions $h_1(x), h_2(x)$, such that $\ell(\boldsymbol{w}; \boldsymbol{x}, y) = -y h_1(\boldsymbol{x}^T \boldsymbol{w}) + h_2(\boldsymbol{x}^T \boldsymbol{w})$.*

It will be convenient to consider population risk directly. Now, we adopt standard setting of learning problems, where each data point $(\boldsymbol{x}, y)$ is drawn from some underlying unknown distribution $\mathcal{D}$ and $\|\boldsymbol{x}\|_2 \leqslant 1$. Given a SGLL $\ell(\boldsymbol{w}; \boldsymbol{x}, y)$, the population loss is defined as $L(\boldsymbol{w}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\boldsymbol{w}; \boldsymbol{x}, y)$. For simplicity, instead of assuming $\boldsymbol{w}$ belongs to $B_2(r)$, we use the following equivalent notation: $\ell(\boldsymbol{w}; \boldsymbol{x}, y) = -yh_1(r\boldsymbol{x}^T\boldsymbol{w}) + h_2(r\boldsymbol{x}^T\boldsymbol{w})$, and the constraint set for $\boldsymbol{w}$ is $\mathcal{C} = B_2(1)$. Denote $G(\boldsymbol{w}; \boldsymbol{x}, y) = \nabla\ell(\boldsymbol{w}; \boldsymbol{x}, y) = rm(\boldsymbol{w}; \boldsymbol{x}, y)\boldsymbol{x}$, where $m(\boldsymbol{w}; \boldsymbol{x}, y) = h_2'(r\boldsymbol{x}^T\boldsymbol{w}) - yh_1'(r\boldsymbol{x}^T\boldsymbol{w})$. Suppose $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\|G(\boldsymbol{w}; \boldsymbol{x}, y) - g(\boldsymbol{w})\|_2^2] \leqslant \sigma_0^2$, where $g(\boldsymbol{w}) = \nabla L(\boldsymbol{w})$. This is a common assumption in stochastic optimization literature, such as (Bubeck et al., 2015).

Given any $\alpha > 0$, we hope to design a noninteractive local DP mechanism with low sample complexity, such that the final output point $\boldsymbol{w}^{priv}$ satisfies $L(\boldsymbol{w}^{priv}) - L(\boldsymbol{w}^*) \leqslant \alpha$.

For GLM loss functions, it is easy to see that the stochastic gradient evaluated on w with data point x_i is at the same direction with x_i. So adding isotropic noise to x_i provides "unbiased" information about direction of stochastic gradient. However, the magnitude is a nonlinear function of $w^T x_i$, making it hard for SGD even to converge to population minimizer. This is why we seek to find polynomial approximation of the magnitude of gradients.

To estimate the magnitude of gradients, we use Chebyshev polynomials to approximate nonlinear univariate function $f_i(x) = h_i'(rx)$, where $x \in [-1, 1]$. For brevity of notations, we just use $f(x)$ to represent either $f_1(x)$ or $f_2(x)$. Denote the Chebyshev approximation with degree p as $\hat{f}_p(x) = \frac{1}{2} + \sum_{k=1}^{p} a_k T_k(x)$, where $T_k(x)$ is the k-th Chebyshev polynomial, and $a_k = \frac{2}{\pi}\int_{-1}^{1}\frac{f(x)T_k(x)}{\sqrt{1-x^2}}\mathrm{d}x$ is the corresponding coefficient. According to existing results about Chebyshev approximations and some calculations, we have the following lemma:

Lemma 8. *Given any $\alpha > 0$, by setting $k = c\ln\frac{1}{\alpha}, p = \lceil k + e\mu_2(k; r)\rceil$, where c is a constant, we have $\left\|\hat{f}_p(x) - f(x)\right\|_\infty \leqslant \alpha$*

The Chebyshev approximations with degree p for $f_i(x)$ $(i = 1, 2)$ are denoted as $\hat{f}_{ip}(x) = \frac{1}{2} + \sum_{k=1}^{p} a_{ik}T_k(x) = \sum_{k=0}^{p} c_{ik}x^k$, where c_{ik} is the coefficient of term x^k. Now we approximate $m(\boldsymbol{w}; \boldsymbol{x}, y)$ and $G(\boldsymbol{w}; \boldsymbol{x}, y)$ as follows:

$$\begin{aligned}\hat{m}(\boldsymbol{w}; \boldsymbol{x}, y) :=& - y\hat{f}_{1p}(r\boldsymbol{x}^T\boldsymbol{w}) + \hat{f}_{2p}(r\boldsymbol{x}^T\boldsymbol{w})\\ =&\sum_{k=0}^{p}(c_{2k} - c_{1k}y)(r\boldsymbol{x}^T\boldsymbol{w})^k\\ \hat{G}(\boldsymbol{w}; \boldsymbol{x}, y) :=& r\hat{m}(\boldsymbol{w}; \boldsymbol{x}, y)\boldsymbol{x}\end{aligned}$$

With these approximations, we state our mechanism in Algorithm 5, where Basic Private Vector mechanism is given in Algorithm 1. Note an important trick in Step 6-8 of Algorithm 5, is that: we run basic private mechanism p times, to obtain fresh private copies of the same vector $\boldsymbol{x}$, which are then used to calculate an unbiased estimation of $\hat{G}(\boldsymbol{w}; \boldsymbol{x}, y)$ with variance as low as possible (i.e. line 8 in Algorithm 6).The LDP property of Algorithm 5 is given as follows: The privacy proof directly follows from Basic Vector Mechanism and Composition Theorem.

Algorithm 5 LDP SGLD Mechanism - Collection

```
Input: Personal data $(\boldsymbol{x}, y)$, expansion order $p$, privacy parameter $\epsilon, \delta$
Output: Private synopsis $b = \{z_{yi}, \boldsymbol{z}_j | i \in \{0\} \cup [p], j \in [p(p+1)/2]\}$ sent to the server
1: Setting $\epsilon_y = \frac{\epsilon}{4(p+1)}, \delta_y = \frac{\delta}{4(p+1)}, \epsilon_1 = \frac{\epsilon}{p(p+1)}, \delta_1 = \frac{\delta}{p(p+1)}$
2: $\boldsymbol{z}_0 \leftarrow$ Basic Private Vector$(\boldsymbol{x}, \epsilon/4, \delta/4)$
3: for $i = 0, 1, \ldots, p$ do
4:     $z_{yj} \leftarrow$ Basic Private Vector$(y, \epsilon_y, \delta_y)$
5: end for
6: for $j = 1, \ldots, \frac{p(p+1)}{2}$ do
7:     $\boldsymbol{z}_j \leftarrow$ Basic Private Vector$(\boldsymbol{x}, \epsilon_1, \delta_1)$
8: end for
```

Theorem 4. *LDP SGLD Mechanism 5 preserves (ϵ, δ)-LDP.*

Having obtained the private synopsis sent by all uers, now the server can construct a stochastic inexact gradient oracle (defined in Defintion 4) for any point $\boldsymbol{w} \in \mathcal{C}$, as stated in Algorithm 6.

Definition 4. *(Dvurechensky & Gasnikov, 2016) For an objective function $f(\boldsymbol{w})$, a (γ, β, σ) stochastic oracle returns a turple $(F_{\gamma,\beta,\sigma}(\boldsymbol{w}; \boldsymbol{\xi}), G_{\gamma,\beta,\sigma}(\boldsymbol{w}; \boldsymbol{\xi}))$, such that:*

$$\begin{gathered}\mathbb{E}_{\boldsymbol{\xi}}[F_{\gamma,\beta,\sigma}(\boldsymbol{w}; \boldsymbol{\xi})] = f_{\gamma,\beta,\sigma}(\boldsymbol{w})\\ \mathbb{E}_{\boldsymbol{\xi}}[G_{\gamma,\beta,\sigma}(\boldsymbol{w}; \boldsymbol{\xi})] = g_{\gamma,\beta,\sigma}(\boldsymbol{w})\\ \mathbb{E}_{\boldsymbol{\xi}}[\|G_{\gamma,\beta,\sigma}(\boldsymbol{w}; \boldsymbol{\xi}) - g_{\gamma,\beta,\sigma}(\boldsymbol{w})\|^2] \leqslant \sigma^2\\ 0 \leqslant h(\boldsymbol{v}, \boldsymbol{w}) \leqslant \frac{\beta}{2}\|\boldsymbol{v} - \boldsymbol{w}\|^2 + \gamma, \forall \boldsymbol{v}, \boldsymbol{w} \in \mathcal{C}\end{gathered}$$

where $h(\boldsymbol{v}, \boldsymbol{w}) = f(\boldsymbol{v}) - f_{\gamma,\beta,\sigma}(\boldsymbol{w}) - \langle g_{\gamma,\beta,\sigma}(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w}\rangle$.

For any $(\boldsymbol{x}, y)$ in the domain, as loss function $\ell(\boldsymbol{w}; \boldsymbol{x}, y)$ is convex and β-smooth with respect to $\boldsymbol{w}$, we can prove the following lemma:

Lemma 9. *For any $\gamma > 0$, setting $k = c\ln\frac{4r}{\gamma}, p = \lceil k + 2\mu_2(k; r)\rceil$, then Algorithm 6 outputs a (γ, β, σ) stochastic oracle defined in Definition 4, where $\sigma = \tilde{O}\left(\sigma_0 + \gamma + \frac{p^{2p+1}(4r)^{p+1}}{\epsilon^{p+2}}\right)$.*

Algorithm 6 LDP SGLD Mechanism - Learning

```
Input: Private synopsis $b = \{z_y, \boldsymbol{z}_j | j \in \{0\} \cup [p(p+1)/2]\}$ of each user, public coefficients $\{c_{1k}, c_{2k} | k \in \{0\} \cup [p]\}$, initial point $\boldsymbol{w}_1$
Output: Learned classifier $\boldsymbol{w}^{priv}$
for $s = 1, \ldots, n$ do
    \\ Construct stochastic inexact gradient
    \\ Denote the private synopsis of user $s$ as $b$ above for abbreviation
    Set $t_0 = 1$
    for $j = 1, \ldots, p$ do
        $t_j = \prod_{i=j(j-1)/2+1}^{j(j+1)/2} (\boldsymbol{w}_s^T \boldsymbol{z}_i)$
    end for
    $\tilde{G}(\boldsymbol{w}_s; b) \leftarrow \left( \sum_{k=0}^{p} (c_{2k} - c_{1k} z_{yj}) t_k r^{k+1} \right) \boldsymbol{z}_0$
    \\ One update via SIGM
    Run one iteration of SIGM algorithm with $\tilde{G}(\boldsymbol{w}_s, b)$ and obtain $\boldsymbol{w}_{s+1}$
end for
Set $\boldsymbol{w}^{priv} := \boldsymbol{w}_{n+1}$
```

Based on above (γ, β, σ) stochastic oracle, and the algorithm proposed in SIGM paper (Dvurechensky & Gasnikov, 2016) (omitted here, due to the limitation of space), our complete learning algorithm is given in Algorithm 6. Before proving our sample complexity, we state the basic convergence result of SIGM algorithm:

Lemma 10 ((Dvurechensky & Gasnikov, 2016)). *Assume a function $f(\boldsymbol{w})$ (suppose constrain set is $\mathcal{W}$) is endowed with a (γ, β, σ) stochastic oracle, then the sequence $\boldsymbol{w}_k$ (corresponds to $\boldsymbol{y}_k$ in the original paper) generated by the SIGM algorithm satisfies:*

$$\mathbb{E}[f(\boldsymbol{w}_k)] - f(\boldsymbol{w}^*) \leqslant O\left(\frac{\sigma}{\sqrt{k}} + \gamma\right)$$

where expectation is over the randomness of the stochastic oracle and $\boldsymbol{w}^* = \operatorname{argmin}_{\boldsymbol{w} \in \mathcal{W}} f(\boldsymbol{w})$.

The accuracy results directly follows from the quality of inexact stochastic gradient oracle we constructed, and the convergence result of SIGM.

Theorem 5. *Consider smooth generalized linear loss. For any setting $\alpha > 0$, by setting $\gamma = \frac{\alpha}{2}, k = c \ln \frac{4r}{\gamma}, p = \lceil k + 2\mu_2(k; r) \rceil$ in Algorithm 5, 6, if*

$$n > O\left((\frac{8r}{\alpha})^{4r \ln \ln(8r/\alpha)} \left(\frac{4r}{\epsilon}\right)^{2cr \ln(8r/\alpha)+2} \left(\frac{1}{\alpha^2 \epsilon^2}\right) \right),$$

we can achieve loss guarantee $L(\boldsymbol{w}^{priv}) - L(\boldsymbol{w}^*) \leqslant \alpha$

As we can see, learning in non-interactive LDP model is more difficult than interactive form, especially when loss is highly nonlinear, we even can not obtain an unbiased estimation either for objective function or gradients. However, our method shows it possible to learn smooth GLM with quasi-polynomial sample complexity.

4.1. Example: Learning Logistic Regression

Either from the view of exponential family generalized linear model or the concrete loss function, it is not difficult to see logistic loss belongs to SGLL. For example, in logistic regression, $\ell(\boldsymbol{w}; \boldsymbol{x}, y) = \log(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}) = -\left(\frac{y}{2}\boldsymbol{w}^T\boldsymbol{x}\right) + \left(\frac{1}{2}\boldsymbol{w}^T\boldsymbol{x} + \ln(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}})\right)$. So we let $h_1(x) = \frac{x}{2}, h_2(x) = \frac{x}{2} + \ln(1+e^{-x})$. As we know logistic loss is convex and β-smooth for some parameter β, and the absolutely smooth property of linear function is obvious, hence once we prove $f(x) = \ln(1 + e^{-x})$ is absolutely smooth, then logistic loss satisfies the definition of SGLL.

Proposition 1. *$f(x) = \ln(1 + e^{-x})$ is absolutely smooth with $\mu_1(k; r) = r\sqrt{4k\pi^3}, \mu_2(k; r) = \frac{rk}{e}$*

Hence, we can use private mechanisms (5,6) to learn logistic regression.

Theorem 6. *Consider Logistic regression problem with $\ell(\boldsymbol{w}; \boldsymbol{x}, y) = \log(1 + \exp(-y\boldsymbol{w}^T\boldsymbol{x}))$ For any $\alpha > 0$, by setting $\gamma = \frac{\alpha}{2}, k = c \ln \frac{4r}{\gamma}, p = \lceil k + 2\mu_2(k; r) \rceil$, if $n > O\left((\frac{8r}{\alpha})^{4r \ln \ln(8r/\alpha)} \left(\frac{4r}{\epsilon}\right)^{2cr \ln(8r/\alpha)+2} \left(\frac{1}{\alpha^2 \epsilon^2}\right) \right)$ in Algorithm 5, 6, we can achieve $L(\boldsymbol{w}^{priv}) - L(\boldsymbol{w}^*) \leqslant \alpha$.*

5. Conclusions

In this paper, we consider how to design efficient algorithms for common learning and estimation problems under non-interactive LDP model. In particular, for sparse linear regression and mean estimation problem, we propose efficient algorithms and prove the polynomial dependence of excess risk or square error over $\log d$ and $\frac{1}{n}$, which is exactly to be expected in high dimensional case. We also extend our methods to nonparametric case and show good bounds for Kernel Ridge Regression.

For more difficult smooth generalized linear loss optimization problems, we use private Chebyshev approximations to estimate gradients of the objective loss, combined with existing inexact gradient descent methods to obtain final outputs. The sample complexity of our mechanism is quasi-polynomial with respect to $\frac{1}{\alpha}$, where α is the desired population excess risk.

An interesting open problem is whether our theoretical guarantees are optimal. If not, how to improve them while preserving the efficiency in non-interactive LDP model. We think these problems are critical to understand LDP in the future.

Acknowledgments

This work was partially supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), NSFC (61573026). We would like to thank the anonymous reviewers for their valuable comments on our paper.

References

Bassily, Raef and Smith, Adam. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 127–135. ACM, 2015.

Bassily, Raef, Smith, Adam, and Thakurta, Abhradeep. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 464–473. IEEE, 2014.

Bubeck, Sébastien et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chaudhuri, K. and Monteleoni, C. Privacy-preserving logistic regression. In *Conference on Neural Information Processing Systems, British Columbia, Canada, December*, pp. 289–296, 2008.

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

Duchi, John, Wainwright, Martin J, and Jordan, Michael I. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural Information Processing Systems*, pp. 1529–1537, 2013a.

Duchi, John, Wainwright, Martin, and Jordan, Michael. Minimax optimal procedures for locally private estimation. *arXiv preprint arXiv:1604.02390*, 2016.

Duchi, John C, Jordan, Michael I, and Wainwright, Martin J. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pp. 429–438. IEEE, 2013b.

Dvurechensky, Pavel and Gasnikov, Alexander. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, pp. 265–284. Springer, New York, USA, 2006.

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

Feldman, Vitaly, Guzmán, Cristóbal, and Vempala, Santosh. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1265–1277. Society for Industrial and Applied Mathematics, 2017.

Geng, Quan and Viswanath, Pramod. The optimal mechanism in differential privacy. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 2371–2375. IEEE, 2014.

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *IEEE Symposium on Foundations of Computer Science*, pp. 61–70, 2010.

Hardt, M., Ligett, K., and Mcsherry, F. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pp. 2339–2347, 2012.

Hsu, Daniel and Sabato, Sivan. Loss minimization and parameter estimation with heavy tails. *Journal of Machine Learning Research*, 17(18):1–40, 2016.

Kairouz, Peter, Oh, Sewoong, and Viswanath, Pramod. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pp. 2879–2887, 2014.

Kairouz, Peter, Oh, Sewoong, and Viswanath, Pramod. The composition theorem for differential privacy. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1376–1385, 2015a.

Kairouz, Peter, Oh, Sewoong, and Viswanath, Pramod. Secure multi-party differential privacy. In *Advances in Neural Information Processing Systems*, pp. 2008–2016, 2015b.

Kairouz, Peter, Bonawitz, Keith, and Ramage, Daniel. Discrete distribution estimation under local privacy. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2436–2444, 2016.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? In *IEEE Symposium on Foundations of Computer Science*, pp. 531–540, 2008.

Kasiviswanathan, Shiva Prasad and Jin, Hongxia. Efficient private empirical risk minimization for high-dimensional learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 488–497, 2016.

Kasiviswanathan, Shiva Prasad, Lee, Homin K, Nissim, Kobbi, Raskhodnikova, Sofya, and Smith, Adam. What can we learn privately? *SIAM Journal on Computing*, 40 (3):793–826, 2011.

Kearns, Michael. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Kifer, Daniel, Smith, Adam, and Thakurta, Abhradeep. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1(41):3–1, 2012.

Lei, J. Differentially private m-estimators. In *Advances in Neural Information Processing Systems*, pp. 361–369, 2011.

Rahimi, Ali, Recht, Benjamin, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, pp. 5, 2007.

Rubinstein, B., Bartlett, P. L., Huang, L., and Taft, N. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):4, 2012.

Smith, A. Privacy-preserving statistical estimation with optimal convergence rates. In *ACM Symposium on Theory of Computing, STOC*, pp. 813–822, 2011.

Smith, Adam and Thakurta, Abhradeep. Differentially private model selection via stability arguments and the robustness of the lasso. *J Mach Learn Res Proc Track*, 30: 819–850, 2013.

Talwar, Kunal, Thakurta, Abhradeep, and Zhang, Li. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pp. 3025–3033, 2015.

Thaler, J., Ullman, J., and Vadhan, S. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, volume 7391, pp. 810–821. 2012.

Vershynin, Roman. Estimation in high dimensions: a geometric perspective. In *Sampling theory, a renaissance*, pp. 3–66. Springer, 2015.

Wang, Y., Fienberg, S. E., and Smola, A. J. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pp. 2493–2502, 2015.

Wang, Z., Jin, C., Fan, K., Zhang, J., Huang, J., Zhong, Y., and Wang, L. Differentially private data releasing for smooth queries. *Journal of Machine Learning Research*, 17(51):1–42, 2016.

Warner, Stanley L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

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

Zhang, Jun, Zhang, Zhenjie, Xiao, Xiaokui, Yang, Yin, and Winslett, Marianne. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.