Efficient Private Empirical Risk Minimization for High-dimensional Learning

Shiva Prasad Kasiviswanathan
Samsung Research America, Mountain View, CA 94043

Hongxia Jin
Samsung Research America, Mountain View, CA 94043

Abstract
Dimensionality reduction is a popular approach for dealing with high dimensional data that leads to substantial computational savings. Random projections are a simple and effective method for universal dimensionality reduction with rigorous theoretical guarantees. In this paper, we theoretically study the problem of differentially private empirical risk minimization in the projected subspace (compressed domain). We ask: is it possible to design differentially private algorithms with small excess risk given access to only projected data? In this paper, we answer this question in affirmative, by showing that for the class of generalized linear functions, given only the projected data and the projection matrix, we can obtain excess risk bounds of $O(w(C)^{2/3}/n^{1/3})$ under $(\epsilon)$-differential privacy, and $O(\sqrt{w(C)/n})$ under $(\epsilon, \delta)$-differential privacy, where $n$ is the sample size and $w(C)$ is the Gaussian width of the parameter space $C$ that we optimize over. A simple consequence of these results is that, for a large class of ERM problems, in the traditional setting (i.e., with access to the original data), under $\epsilon$-differential privacy, we improve the worst-case risk bounds of (Bassily et al., 2014).

1. Introduction

Curse of dimensionality is a well-established obstacle in machine learning affecting aspects such as the accuracy, computation time, storage space, and communication cost of many common tasks. Therefore, a general strategy for tackling many high-dimensional learning tasks is first transforming from the data domain to some appropriate compressed measurement domain, and then performing the learning task in the measurement domain. This idea has led to the development of the field of compressed learning (Eladar & Kutyniok, 2012), where the general goal is to perform a machine learning task in the compressed measurement domain with almost the same performance guarantee as that achievable in the original data domain. Efficiency gains achievable through compressed learning have been well documented for variety of common machine learning tasks (Arriaga & Vempala, 2006; Davenport et al., 2006; Maillard & Munos, 2009; Fard et al., 2012; Kabón, 2014; Calderbank et al., 2009; Wright et al., 2009). In addition to the efficiency gains, these techniques also work successfully in situations where one cannot observe the data domain because collecting data might be difficult or expensive. In this case, by learning in the compressed domain, one avoids the cost of recovering back the data in the high-dimensional domain.

A common approach for dimensionality reduction (compression) is using the technique of random projections (Vempala, 2005; Mahoney, 2011), where the original high-dimensional data is projected onto a lower-dimensional subspace using some appropriately chosen random matrix. Random projections, such as the popular Johnson-Lindenstrauss transform and its variants, preserve the structure of the original data space, and hence can be used as a way to reduce the cost of the learning process perceptibly, while preserving the performance approximately. For high-dimensional data, a random projection can lead to substantial savings in resources such as storage space, transmission bandwidth, and processing time.

Machine learning algorithms are frequently run on sensitive data, and this has motivated the study of learning algorithms that have good performance guarantees while satisfying a rigorous notion of privacy called differential privacy (Dwork et al., 2006b). There currently exist differentially private algorithms for many statistical and machine-learning tasks such as classification, regression, PCA, clustering, density estimation, among others. We refer the reader to recent surveys by Sarwate and Chaudhuri (Sarwate & Chaudhuri, 2013), and by Dwork and Roth (Dwork & Roth, 2013) for an overview of recent developments in machine learning with differential privacy.

In this paper, we theoretically study private compressed learning in the framework of Empirical Risk Minimization (ERM). In the traditional (uncompressed) empirical risk minimization framework, we are given $n$ datapoints...
In the empirical risk minimization setting, the success of empirical risk also translates into a bound on the coin tosses of the algorithm; the excess empirical risk is guaranteed that with high probability (or expectation over \$z\$) the algorithm gets access to the true data, private convex ERM mechanisms have been investigated extensively in the literature under both \$\epsilon\$ and \$(\epsilon, \delta)\$-differential privacy notions. Starting with the results of Chaudhuri et al. (Chaudhuri & Monteleoni, 2009; Chaudhuri et al., 2011) private convex ERM problems have been studied in various settings including the low-dimensional setting (Rubinstein et al., 2009; Kifer et al., 2012), high-dimensional sparse regression setting (Kifer et al., 2012; Smith & Thakurta, 2013), online learning setting (Jain et al., 2012; Thakurta & Smith, 2013; Jain & Thakurta, 2014; Mishra & Thakurta, 2015), local privacy setting (Duchi et al., 2013), interactive setting (Jain & Thakurta, 2013; Ullman, 2015), and streaming setting (Kasiviswanathan et al., 2016; Bassily et al., 2014) showed that for a general convex loss function \(f(\theta; z)\) that for every \(z\) is 1-Lipschitz the expected excess empirical risk is at most \(O(\sqrt{d\ell/n})\) under \((\epsilon, \delta)\)-differential privacy and \(O(d/n)\) under \(\epsilon\)-differential privacy (ignoring the dependence on other parameters for simplicity).

In the traditional (uncompressed) setting, when the algorithm gets access to the true data, private convex ERM problems have been studied in various settings including the low-dimensional setting (Rubinstein et al., 2009; Kifer et al., 2012), high-dimensional sparse regression setting (Kifer et al., 2012; Smith & Thakurta, 2013), online learning setting (Jain et al., 2012; Thakurta & Smith, 2013; Jain & Thakurta, 2014; Mishra & Thakurta, 2015), local privacy setting (Duchi et al., 2013), interactive setting (Jain & Thakurta, 2013; Ullman, 2015), and streaming setting (Kasiviswanathan et al., 2016; Bassily et al., 2014) showed that for a general convex loss function \(f(\theta; z)\) that for every \(z\) is 1-Lipschitz the expected excess empirical risk is at most \(O(\sqrt{d\ell/n})\) under \((\epsilon, \delta)\)-differential privacy and \(O(d/n)\) under \(\epsilon\)-differential privacy (ignoring the dependence on other parameters for simplicity).

### Compressed Learning Problem Formulation

Let \(\Phi \in \mathbb{R}^{n \times d}\) be a random projection matrix. In this paper, we use the popular and wide class of subgaussian matrices for random projection. We define the measurement domain \(M\) as:
\[
M = \{ (\Phi x, y) : (x, y) \in Z \}.
\]
In other words, \(M\) is a compressed representation of \(Z\). Our goal is to output an estimator (while satisfying the constraints of differential privacy) that minimizes the empirical risk \((2)\) given access to only the compressed representation of \((x_1, y_1), \ldots, (x_n, y_n)\) (i.e., \((\Phi x_1, y_1), \ldots, (\Phi x_n, y_n)\)) and \(\Phi\). This compressed setting is a strict generalization of the traditional setting where the algorithm gets access to \((x_1, y_1), \ldots, (x_n, y_n)\). Our results also trivially hold for the traditional setting, as given \((x_1, y_1), \ldots, (x_n, y_n)\), we can pick \(\Phi\) and generate the input \((\Phi x_1, y_1), \ldots, (\Phi x_n, y_n)\) for our proposed algorithm.

---

1. Throughout this paper, we measure Lipschitzness with respect to \(L_2\)-norm in the input space.

2. Better bounds can be achieved for strongly convex loss functions (Bassily et al., 2014; Talwar et al., 2015a).

3. In general, given \(\Phi x\), it is not possible to reconstruct \(x\) without further assumptions on \(x\).
Efficient Private Empirical Risk Minimization for High-dimensional Learning

<table>
<thead>
<tr>
<th>Problem Setting</th>
<th>$\epsilon$-differential privacy</th>
<th>$(\epsilon, \delta)$-differential privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed Learning (under random projections)</td>
<td>$\tilde{O}\left(\frac{d^{1/3}3\lambda w(C)^{2/3}/|C|_2}{(en)^{2/3}}\right)$</td>
<td>$\tilde{O}\left(\frac{w\lambda w(C)|C|_2\log^2(1/\delta)}{\sqrt{n}}\right)$</td>
</tr>
<tr>
<td>Traditional (uncompressed) Learning</td>
<td>$\tilde{O}\left(\frac{\lambda w(C)^{2/3}/|C|_2}{(en)^{2/3}}\right)$</td>
<td>$\tilde{O}\left(\frac{\Gamma_0^{1/3}(\lambda w(C)^{2/3}/|C|_2)\log^2(1/\delta)}{(en)^{2/3}}\right)$ (Talwar et al., 2015b)</td>
</tr>
</tbody>
</table>

Table 1. Upper bounds on the excess risk under differential privacy for the convex ERM problem: $\min_{\theta \in C} \frac{1}{n} \sum_{i=1}^{n} f((x_i, \theta); y_i)$ for $\lambda_1$-Lipschitz convex $f$ and general convex set $C$. Here $w(C)$ is the Gaussian width of $C$, $\|C\|_2$ is the diameter of $C$, $\psi$ is the subgaussian norm of random vectors used in the projection matrix, and $\Gamma_0$ is the curvature constant of $f$. For traditional (uncompressed) learning, under $(\epsilon, \delta)$-differential privacy, better bounds can be obtained for some specific $C$’s (Talwar et al., 2015a,b). All the uncited results appear in this paper.

Our Contributions. We primarily make two contributions in this paper. A summary of our results appear in Table 1. All the presented algorithms run in time polynomial in $n$ and $d$.

(a) Compressed Learning. A natural first question is whether non-trivial⁴ private ERM algorithms exist that operate only with compressed data. In this paper, we answer this question in affirmative. Using techniques from convex geometry and high-dimensional estimation, we present a generic mechanism that transforms any differentially private ERM algorithm to provide excess risk bounds from compressed data. The idea is to add noise for privacy in the compressed domain, and then “lift” the result back to the original domain. Our analysis is based on exploiting the geometric structure of $C$. The geometric parameter, Gaussian width, defined as $w(C) = \mathbb{E}_{g \sim N(0,1)^d}[\sup_{\theta \in C} \langle \theta, g \rangle]^2$ plays an important role in our analysis, and shows up repeatedly as a geometric measure of the size of the set $C$. For the important class of generalized linear functions, using the private ERM algorithms from (Bassily et al., 2014), we obtain an excess empirical risk bound of $\approx \frac{\|C\|^{2/3}}{n^{1/3}}$ for achieving $\epsilon$-differential privacy, and $\approx \sqrt{\frac{w(C)}{n}}$ for achieving $(\epsilon, \delta)$-differential privacy.

(b) Traditional Learning (under $\epsilon$-differential privacy). Our bounds on compressed learning also directly translate to the traditional ERM setting. We obtain the first excess risk bounds, under $\epsilon$-differential privacy, that go beyond the worst-case bounds of $O(d/n)$ from (Bassily et al., 2014). The techniques presented by (Talwar et al., 2015a,b) provide only $(\epsilon, \delta)$-differential privacy guarantees (i.e., $\delta$ has to be greater than 0). There are many simple consequences of our result. For example, we obtain an $\epsilon$-differentially private Lasso algorithm with excess risk $\tilde{O}(1/n^{1/3})$, assuming all the input datapoints have bounded $L_2$-norm. Previous best bounds here (from (Bassily et al., 2014)) had a polynomial dependence on the dimension.

Other Related Work. (Jain & Thakurta, 2014) gave dimension-independent expected excess risk bounds for the case generalized linear functions with a strongly convex regularizer and assuming that $C = \mathbb{R}^d$ (unconstrained optimization). Note that in this paper, we do not make any strong convexity assumptions and our results hold for the more interesting constrained optimization. Kifer et al. (Kifer et al., 2012), and Smith and Thakurta (Smith & Thakurta, 2013) studied the problem of releasing Lasso estimator privately under certain assumptions about the instance (restricted strong convexity and mutual incoherence). Under these assumptions, they obtain excess risk of $O(\text{polylog}(d)/n)$. However, these assumptions on the data are rather strong and may not hold in practice (Talwar et al., 2015a,b). Without these data dependent assumptions, (Talwar et al., 2015a,b) present an $(\epsilon, \delta)$-differentially private Lasso algorithm with excess risk of $\tilde{O}(\log(d)/n^{1/3})$. However, the question of designing a private Lasso estimator under $\epsilon$-differential privacy whose excess risk grows logarithmically in the dimension size $d$ was still open, which we resolve here. We perform a more detailed comparison with the results of Talwar et al. in Section 4.

Random projections have been used as a tool to design (differentially) private algorithms in many other problem settings. Blocki et al. (Blocki et al., 2012) have shown that if $\Phi$ is a Gaussian Johnson-Lindenstrauss matrix of appropriate dimension, then $\Phi X$ is differentially private if the least singular value of $X$ is “sufficiently” large. Here $X \in \mathbb{R}^{n \times d}$ is a data matrix of $n$ points in $d$ dimensions. Note that $\Phi X$ does not reduce the dimensionality of the data but rather generates a set of fewer datapoints. The bound on the least singular value was recently improved by (Sheffet, 2015). Wang et al. (Wang et al., 2015) show that for the problem of subspace clustering achieving, random projections can be helpful for both computational efficiency and privacy protection. Kenthapadi et al. (Kenthapadi et al., 2013) use Johnson-Lindenstrauss transform to publish a private sketch that enables estimation of the distance between users. Zhou et al. (Zhou et al., 2009b) provide a technique of generating synthetic data using random linear
or affine transformations. Random projections have also been used to achieve various other weaker privacy guarantees (Duncan et al., 1991; Zhou et al., 2009a).

2. Preliminaries

Notation and Data Normalization. We denote \([n] = \{1, \ldots, n\}\). Vectors are in column-wise fashion, denoted by boldface letters. For a vector \(v\), \(|v|_p\), for \((1 \leq p \leq \infty)\) denotes its \(L_p\)-norm. For \(p = 2\), we drop the subscript and use \(|v|\) to denote its Euclidean \((L_2)\) norm. The \(d\)-dimensional unit ball in \(L_p\)-norm centered at origin is denoted by \(B^d_p\), and the \(d\)-dimensional Euclidean unit sphere is denoted by \(S^{d-1}\). For a variable \(n\), we use \(poly(n)\) to denote a polynomial function of \(n\) and \(\text{poly}(\log(n))\) to denote \(\text{polynomial} + \text{poly}(\log(n))\).

Throughout this paper, we assume that all the inputs are \(L_2\)-normalized with \(|x| \leq 1\) and \(|y| \leq 1\). This is for simplicity, and our results extend to the case without this normalization (with probably an increased Lipschitz constant). Also as is typical in random projection based analyses, we assume that the \(x_i\)’s are selected independent of the projection matrix \(\Phi\).

We refer the reader to (Boyd & Vandenberghe, 2004) for standard definitions in convex optimization. For a set of vectors, we define its diameter as the maximum attained \(L_2\)-norm in the set.

Definition 1. (Diameter of Set) The diameter of a closed set \(C \subseteq \mathbb{R}^d\), is defined as \(\|C\|_2 = \sup_{\theta \in C} |\theta|\).

Definition 2. (Lipschitz Functions over \(\theta\)) A loss function \(f : C \times Z \to \mathbb{R}\) is \(\lambda_f\)-Lipschitz with respect to \(\theta\) over the domain \(C\), if for any \(z \in Z\), and \(\theta_a, \theta_b \in C\), we have \(|f(\theta_a; z) - f(\theta_b; z)| \leq \lambda_f |\theta_a - \theta_b|\).

For a generalized linear function \(\ell(x; \theta; y)\), let \(\lambda_\ell\) denote the Lipschitz constant of \(\ell\) in the first argument. The following claim easily follows from our normalization (proof in Appendix A, Supplementary material).

Claim 2.1. Let function \(\ell(x; \theta; y)\) be \(\lambda_\ell\)-Lipschitz in the first argument over the domain \(\mathbb{R}\). Then \(\ell\) is \(\lambda_\ell\)-Lipschitz with respect to \(\theta\).

For common loss functions, \(\lambda_{\ell}\) is small. For squared loss, used in linear regression, the loss function is \(2\|\cdot\|_2\) Lipschitz; for logistic loss, used in logistic regression, the loss function is 1-Lipschitz; for hinge loss, used in SVM, the loss function is 1-Lipschitz.

Subgaussian matrices are a popular choice for random projections (matrix \(\Phi\)).

Definition 3 (Subgaussian Random Variable and Vector). We call a random variable \(a \in \mathbb{R}\) subgaussian if there exists a constant \(C > 0\) if \(\Pr(|a| > t) \leq 2 \exp(-t^2/C^2)\) for all \(t > 0\). We say that a random vector \(a \in \mathbb{R}^d\) is subgaussian if the one-dimensional marginals \((a, b)\) are subgaussian random variables for all \(b \in \mathbb{R}^d\).

The class of subgaussian random variables includes many random variables that arise naturally in data analysis, such as Gaussian, Bernoulli, spherical, bounded (where the random variable \(a\) satisfies \(|a| \leq M\) almost surely for some fixed \(M\)). The natural generalizations of these random variables to higher dimension are all subgaussian random vectors. For many isotropic convex sets \(\mathcal{K}\) (such as the hypersphere), a random vector \(a\) uniformly distributed in \(\mathcal{K}\) is subgaussian.

Definition 4 (Norm of Subgaussian Random Variable and Vector). The \(\psi_2\)-norm of a subgaussian random variable \(a \in \mathbb{R}\), denoted by \(\|a\|_{\psi_2}\), is:

\[
\|a\|_{\psi_2} = \inf\left\{t > 0 : \mathbb{E}[\exp(|a|^2/t^2)] \leq 2\right\}.
\]

The \(\psi_2\)-norm of a subgaussian random vector \(a \in \mathbb{R}^d\) is:

\[
\|a\|_{\psi_2} = \sup_{b \in S^{d-1}} \|(a, b)\|_{\psi_2}.
\]

Our analysis is based on exploiting the geometric properties of \(C\). We use the well-studied quantity of Gaussian width that captures the \(L_2\)-geometric complexity of \(C\).

Definition 5 (Gaussian Width). Given a closed set \(S \subseteq \mathbb{R}^d\), its Gaussian width \(w(S)\) is defined as:

\[
w(S) = \mathbb{E}_{a \in S \setminus N(0, 1)^d} \left[\|a, g\|\right].
\]

In particular, \(w(S)^2\) can be thought as the “effective dimension” of \(S\). Many popular convex sets have low Gaussian width, e.g., the width of both the unit \(L_1\)-ball in \(\mathbb{R}^d\) and the standard \(d\)-dimensional probability simplex are both \(\Theta(\sqrt{\log d})\), and the width of any ball \(B^d_2\) for \(1 \leq p \leq \infty\) is \(\approx d^{1-1/p}\). For a set \(C\) contained in the \(B^d_2\), \(w(C)\) is always \(O(\sqrt{d})\).

Differential Privacy Background. Differential privacy is a rigorous notion of privacy that emerged from a long line of work in theoretical computer science (Dwork et al., 2006b). We say two datasets \(D\) and \(D'\) of size \(n\) are neighbors if they differ in one entry.

Definition 6. (Dwork et al., 2006b;a) A randomized algorithm \(A\) is \((\epsilon, \delta)\)-differentially private if for all neighboring datasets \(D, D'\) and for all events \(R \subseteq \mathbb{R}\) in the output space of \(A\), we have \(\Pr[A(\mathbb{R}) \in R] \leq \exp(\epsilon) \cdot \Pr[A(\mathbb{R'}) \in R] + \delta\), where the probability is taken over the randomness of the algorithm. When \(\delta = 0\), the Algorithm is \(\epsilon\)-differentially private.

3. Algorithm for the Compressed Setting

In this section, we present a generic mechanism (Mechanism PROjERM) for private risk minimization on compressed data. Instantiating the generic mechanism with
recent differentially private algorithms of (Bassily et al., 2014) provides our main upper bounds (Theorem 3.11). Remember that in the compressed setting the algorithm gets access (to only) \((\Phi x_i, y_i)\) and the matrix \(\Phi\), and the goal of the algorithm is to privately output an estimator that minimizes the empirical risk \((2)\). In the next section, we discuss a simple extension of our results to the traditional (uncompressed) setting and its consequences. Missing proofs from this section are presented in Appendix B (Supplementary material).

We focus on a large class of optimization problems expressed as a sum of a generalized linear loss function over individual datapoints.

In the following, let \(\hat{\theta}\) be the true minimizer of \(\mathcal{L}(\theta; (x_1, y_1), \ldots, (x_n, y_n)) = \frac{1}{n} \sum_{i=1}^{n} \ell((x_i, \theta); y_i), \) i.e.,

\[
\hat{\theta} \in \arg\min_{\theta \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \ell((x_i, \theta); y_i).
\]

As a simple corollary to the above theorem it also follows that,

**Corollary 3.2.** Under the setting of Theorem 3.1, there exists a constant \(C' > 0\) such that for any \(0 < \gamma, \beta < 1\)

\[
\Pr \left[ \sup_{a, b \in \mathcal{S}} |\langle \Phi a, \Phi b \rangle - \langle a, b \rangle| \geq \gamma \|a\| \|b\| \right] \leq \beta,
\]

provided that \(m \geq \frac{C' w^4}{\gamma^2} \max \left\{ w(S)^2, \log \left( \frac{1}{\beta} \right) \right\} \).

Throughout the rest of this paper, we are going to assume that \(\gamma < 1\) (\(\gamma\) in fact will be set much smaller). Consider the following modified version of the optimization objective from \((3)\),

\[
\mathcal{L}_{\text{comp}}(\theta; (x_1, y_1), \ldots, (x_n, y_n); \Phi) = \frac{1}{n} \sum_{i=1}^{n} \ell((\Phi x_i, \Phi \theta); y_i).
\]

Since in the compressed setting, we have access to \((\Phi x_i, y_i)'s\), we can solve \((4)\). Our idea will be to privately minimize \((4)\) problem over the domain \(\Phi \mathcal{C}\), and use the result to bound the excess risk. Let \(\Phi \mathcal{C} = \{ \theta \in \Phi \theta : \theta \in \mathcal{C} \}\). Note for a convex \(\mathcal{C}\), \(\Phi \mathcal{C}\) is also convex.

Let \(\lambda_{\mathcal{L}_{\text{comp}}}\) denote the Lipschitz constant of the function \(\ell((\Phi x, \theta); y)\) with respect to \(\theta\) over the domain \(\Phi \mathcal{C}\):

\[
\lambda_{\mathcal{L}_{\text{comp}}} = \sup_{(x, y) \in (x_1, y_1), \ldots, (x_n, y_n)} \frac{|\ell((\Phi x, \theta_a); y) - \ell((\Phi x, \theta_b); y)|}{\|\theta_a - \theta_b\|}.
\]

Note that in the above definition of \(\lambda_{\mathcal{L}_{\text{comp}}}\), \((x, y)\) is restricted to any fixed collection of \(n\) elements. The following lemma shows for large enough \(m\), with probability at least \(1 - \beta\), that \(\lambda_{\mathcal{L}_{\text{comp}}}\) is at most \(2\lambda_\ell\). Remember that the Lipschitz constant of \(\mathcal{L}\) is at most \(\lambda_\ell\).

**Lemma 3.3.** Let \(\Phi\) be a random matrix as defined in Theorem 3.1 with \(m = \Theta((w^4 / \gamma^2) \log(n/\beta))\) for \(\beta > 0\). Then with probability at least \(1 - \beta\), the Lipschitz constant of \(\mathcal{L}_{\text{comp}}(\lambda_{\mathcal{L}_{\text{comp}}})\) is at most \(2\lambda_\ell\) with respect to \(\theta\) over the domain \(\Phi \mathcal{C}\).

The following lemma follows from Lipschitz properties of \(\ell\) and Theorem 3.1.

**Lemma 3.4.** Let \(\Phi\) be a random matrix as defined in Theorem 3.1 with \(m = \Theta((w^4 / \gamma^2) \log(n/\beta))\) for \(\beta > 0\). Then with probability at least \(1 - \beta\), for every \((x_i, y_i)\)

\[
\ell((\Phi x_i, \Phi \theta); y_i) \leq \ell((x_i, \hat{\theta}); y_i) + \lambda \gamma \|C\|_2.
\]

From Lemma 3.4, the following lemma is immediate.

**Lemma 3.5.** Let \(\Phi\) be a random matrix as defined in Theorem 3.1 with \(m = \Theta((w^4 / \gamma^2) \log(n/\beta))\) for \(\beta > 0\). Then
Mechanism 1 PROJERM

Input: A random subgaussian matrix $\Phi \in \mathbb{R}^{m \times d}$, and a dataset $D = (\Phi x_1, y_1), \ldots, (\Phi x_n, y_n)$ of $n$ datapoints from the domain $\mathcal{M}_\Phi = \{(\Phi x, y) : x \in \mathbb{R}^d, \|x\| \leq 1, y \in \mathbb{R}, |y| \leq 1\}$

Output: $\theta^{\text{priv}}$ a differentially private estimate of $\theta \in \text{argmin}_{\theta \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^n \ell((\Phi x_i, \theta); y_i)$

1. Let $\theta^{\text{priv}} \leftarrow \text{Output of an } (\epsilon, \delta)\text{-differentially private or an } \epsilon\text{-differentially private ERM algorithm solving the following problem:}$

$$\min_{\theta \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^n \ell((\Phi x_i, \theta); y_i)$$

with probability at least $1 - \beta$,

$$\frac{1}{n} \sum_{i=1}^n \ell((\Phi x_i, \Phi \theta); y_i) \leq \frac{1}{n} \sum_{i=1}^n \ell((\Phi x_i, \hat{\theta}); y_i) + \lambda \epsilon \gamma \|C\|_2.$$

We now present a generic mechanism (Mechanism PROJERM) for privately releasing M-estimator based on minimizing (4) under differential privacy. The mechanism is simple: it “solves” the private ERM problem in the projected subspace, and then “lifts” back the result to the original domain by solving a linear estimation problem. As we will see, working in the compressed domain has the added advantage that the noise added for privacy scales as $\approx \sqrt{m}$.

Mechanism PROJERM is $(\epsilon, \delta)$- or $\epsilon$-differentially private based on the algorithm used in Step 1. Post-processing of the output does not affect the differential privacy guarantee. Also note that Mechanism PROJERM is differentially private independent of the choice of $\Phi$. However, the utility guarantee of Mechanism PROJERM (Theorem 3.1) will require $\Phi$ to be a subgaussian matrix (as in Theorem 3.1) with a lower bound on $m$ that will depend among various parameters, the geometry of $\mathcal{C}$.

In the Step 1 of Mechanism PROJERM any $(\epsilon, \delta)$- or $\epsilon$-differentially private ERM algorithm can be used. In Proposition 3.7, we provide bounds obtained by using the differentially private ERM algorithms of (Bassily et al., 2014). Their $(\epsilon, \delta)$-differentially private ERM algorithm is based on a noisy stochastic variant of the classic gradient descent algorithm. While their $\epsilon$-differentially private ERM algorithm is based on a polynomial time implementation of the exponential mechanism of McSherry and Talwar (McSherry & Talwar, 2007).

Theorem 3.6 (Bassily et al., 2014, Theorem 2.4). Let $f(\theta; z)$ be a convex loss function that is $\lambda$-Lipschitz with respect to $\theta$ over the domain $\mathcal{C} \subseteq \mathbb{R}^d$.

1. There exists an $(\epsilon, \delta)$-differentially private algorithm $A_{\epsilon, \delta}$ that on an input dataset $z_1, \ldots, z_n$ outputs $\hat{\theta} \in \mathcal{C}$ such that for any $\beta > 0$, with probability at least $1 - \beta$,

$$\frac{1}{n} \sum_{i=1}^n f(\hat{\theta}; z_i) - \min_{\theta \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^n f(\theta; z_i)$$

$$= O\left(\frac{\lambda \epsilon \sqrt{d} \|C\|_2 \log^{3/2}(n/\delta) \sqrt{\log(1/\delta) \text{polylog}(1/\beta)}}{\epsilon \gamma \|C\|_2}.\right)$$

2. There exists an $\epsilon$-differentially private algorithm $A$, that on an input dataset $z_1, \ldots, z_n$ outputs $\hat{\theta} \in \mathcal{C}$ such that for any $\beta > 0$, with probability at least $1 - \beta$,

$$\frac{1}{n} \sum_{i=1}^n f(\hat{\theta}; z_i) - \min_{\theta \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^n f(\theta; z_i)$$

$$= O\left(\frac{\lambda d \|C\|_2 \text{polylog}(1/\beta)}{\epsilon \gamma \|C\|_2}.\right)$$

The following proposition follows by combining Lemma 3.5 with Theorem 3.6. The proposition shows that, even for a small projected dimension $m$, the value of the function $L_{\text{comp}}(\theta; (x_1, y_1), \ldots, (x_n, y_n); \Phi)$ at $\theta = \theta^{\text{priv}}$ approximates the minimum empirical risk.

Proposition 3.7. Let $\Phi$ be a random matrix as defined in Theorem 3.1 with $m = \Theta((\psi^2 / \gamma^2) \log(n/\beta))$ for $\beta > 0$. Then the output $\theta^{\text{priv}}$ of Mechanism PROJERM,

1. When invoked (in Step 1) with Algorithm $A_{\epsilon, \delta}$ from Theorem 3.6, with probability at least $1 - \beta$ satisfies:

$$L_{\text{comp}}(\theta^{\text{priv}}; (x_1, y_1), \ldots, (x_n, y_n); \Phi) - L(\hat{\theta}; (x_1, y_1), \ldots, (x_n, y_n))$$

$$= O\left(\frac{\lambda L_{\text{comp}} \sqrt{m} \|C\|_2 \log^{3/2}(n/\delta) \sqrt{\log(1/\delta) \text{polylog}(1/\beta)}}{\epsilon \gamma \|C\|_2}.\right)$$

2. When invoked (in Step 1) with Algorithm $A$, from Theorem 3.6, with probability at least $1 - \beta$ satisfies:

$$L_{\text{comp}}(\theta^{\text{priv}}; (x_1, y_1), \ldots, (x_n, y_n); \Phi) - L(\hat{\theta}; (x_1, y_1), \ldots, (x_n, y_n))$$

$$= O\left(\frac{\lambda L_{\text{comp}} m \|C\|_2 \text{polylog}(1/\beta)}{\epsilon \gamma \|C\|_2}.\right) + \lambda \epsilon \|C\|_2.$$
Existence of $\theta^{\text{priv}}$. To lift $\theta^{\text{priv}}$ into the original $d$-dimensional space, we use recent results about reconstruction from linear observations (Vershynin, 2014). Since $\theta^{\text{priv}} \in \Phi C$, we know that there exists a $\theta^{\text{true}} \in C$, such that $\Phi \theta^{\text{true}} = \theta^{\text{priv}}$. Then the goal is to estimate $\theta^{\text{true}}$ from $\theta^{\text{priv}}$. Again geometry of $C$ (Gaussian width) plays an important role, as it controls the diameter of high-dimensional random sections of $C$ (referred to as $M^*$ bound (Ledoux & Talagrand, 2013; Vershynin, 2014)). We refer the reader to the excellent tutorial by Vershynin (Vershynin, 2014) for more details.

We define Minkowski functional, as commonly used in geometric functional analysis and convex analysis.

**Definition 7** (Minkowski functional). For any vector $\theta \in \mathbb{R}^d$, the Minkowski functional of $C$ (a closed set) is the non-negative number $\|\theta\|_C$ defined by the rule: $\|\theta\|_C = \inf \{ \tau \in \mathbb{R} : \theta \in \tau C \}$.

For the typical situation in ERM problems, where $C$ is a symmetric convex body, then $\|\cdot\|_C$ defines a norm.

The optimization problem solved in Step 2 of Mechanism PROJERM is convex if $C$ is convex, and in fact is a linear program is $C$ is a polytope, and hence can be efficiently solved. The existence of $\theta^{\text{priv}}$ follows from Theorem 3.8, which in fact can be used to bound the distance between $\theta^{\text{priv}}$ and $\theta^{\text{true}}$ (Corollary 3.9).

**Theorem 3.8.** (Vershynin, 2014; Mendelson et al., 2007) Let $\Phi$ be an $m \times d$ matrix, whose rows $\phi_1, \ldots, \phi_m$ are i.i.d., mean zero, isotropic and subgaussian random vectors in $\mathbb{R}^d$ with $\psi = \|\phi_1\|_2$. Let $C$ be a convex set. Given $v = \Phi u$ and $\Phi$, let $\bar{u}$ be the solution to the following convex program: $\min_{u \in \mathbb{R}^d} \|u\|_C$ subject to $\Phi^T u = v$. Then for any $\beta > 0$, with probability at least $1 - \beta$,

$$
\sup_{u, v = \Phi u} \|u - \bar{u}\|_C = O \left( \frac{\psi^4 w(C)}{\sqrt{m}} + \frac{\psi^3 \|C\|_2 \sqrt{\log(1/\beta)}}{\sqrt{m}} \right).
$$

**Corollary 3.9.** If $\Phi$ is a subgaussian matrix (as in Theorem 3.1), then for any $\beta > 0$, with probability at least $1 - \beta$, $\|\theta^{\text{true}} - \theta^{\text{priv}}\|_C = O \left( \frac{\psi^4 w(C)}{\sqrt{m}} + \frac{\psi^3 \|C\|_2 \sqrt{\log(1/\beta)}}{\sqrt{m}} \right)$.

One last thing to be verified is that $\theta^{\text{priv}}$ generated by Mechanism PROJERM is in $C$. This is simple as by definition of Minkowski functional, as for any closed set $C = \{ \theta \in \mathbb{R}^d : \|\theta\|_C \leq 1 \}$. Hence, $\|\theta^{\text{true}}\|_C \leq 1$. By choice of $\theta^{\text{priv}}$ in Step 2, ensures that $\|\theta^{\text{priv}}\|_C \leq \|\theta^{\text{true}}\|_C \leq 1$, which (by definition) shows that $\theta^{\text{true}} \in C$.

Now that we have established that $\theta^{\text{true}} \in C$ exists, we lower bound $\mathcal{L}_{\text{comp}}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n); \Phi)$ in terms of $\mathcal{L}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n))$. Later we will use this lower bound along with Proposition 3.7 to bound the excess empirical risk.

**Lemma 3.10.** Let $\Phi$ be a random matrix as defined in Theorem 3.1 with $m = \Theta((\psi^4/\gamma^2)(w(C) + \sqrt{\log n})^2 \log(n/\beta))$ for $\beta > 0$. Then with probability at least $1 - \beta$,

$$
\mathcal{L}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n))
\leq \mathcal{L}_{\text{comp}}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n); \Phi) + \lambda \gamma \|C\|_2.
$$

Using Lemma 3.10 and Proposition 3.7 provides the following bounds on excess empirical risk. The parameter $\gamma$ is chosen to balance various opposing factors. We will assume that $\gamma = o(1)$, which in the following theorem happens when $w(C) \ll n$ (for achieving $(\epsilon, \delta)$-differential privacy) and $w(C) \ll \sqrt{n}$ (for achieving $\epsilon$-differential privacy). The running time for both these cases is polynomial in $n$ and $m$.

**Theorem 3.11** (Utility of Mechanism PROJERM). Let $\Phi$ be a random matrix as defined in Theorem 3.1.

1. Then the output $\theta^{\text{priv}}$ of Mechanism PROJERM when invoked with Algorithm $\mathcal{A}_{\epsilon, \delta}$ from Theorem 3.6 and $m = \Theta \left( \frac{\psi^4 w(C) + \sqrt{\log n}}{\sqrt{\log(n/\beta)}} \right)$ for $\beta > 0$, with probability at least $1 - \beta$ satisfies:

$$
\mathcal{L}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n)) - \mathcal{L}(\hat{\theta}; (x_1, y_1), \ldots, (x_n, y_n))
= O \left( \frac{\psi \sqrt{w(C)} \lambda \|C\|_2 \log^2 n \log^2(\frac{1}{\beta}) \text{polylog}(\frac{1}{\beta})}{\sqrt{\epsilon n}} \right).
$$

2. Then the output $\theta^{\text{priv}}$ of Mechanism PROJERM when invoked with Algorithm $\mathcal{A}_{\epsilon, \delta}$ from Theorem 3.6 and $m = \Theta \left( \frac{w(C)^{3/2} (w(C) + \sqrt{\log n})^2 \log(n/\beta)}{w(C)^{3/2}} \right)$ for $\beta > 0$, with probability at least $1 - \beta$ satisfies:

$$
\mathcal{L}(\theta^{\text{true}}; (x_1, y_1), \ldots, (x_n, y_n)) - \mathcal{L}(\hat{\theta}; (x_1, y_1), \ldots, (x_n, y_n))
= O \left( \frac{\psi^{4/3} w(C)^{2/3} \lambda \|C\|_2 \log^2 n \text{polylog}(\frac{1}{\beta})}{(\epsilon n)^{1/3}} \right).
$$

**4. Algorithm for the Traditional Setting**

Mechanism PROJERM can also be directly utilized for solving the traditional private empirical risk minimization problem, where the algorithm gets access to $(x_1, y_1), \ldots, (x_n, y_n)$. This approach is illustrated in Mechanism STDERM.

Mechanism STDERM is again $(\epsilon, \delta)$- or $\epsilon$-differentially private based on how Mechanism PROJERM is invoked. The excess empirical risk bound of Mechanism STDERM follows from Theorem 3.11.

**Corollary 4.1** (of Theorem 3.11). The output $\theta^{\text{priv}}$ of Mechanism STDERM has the same privacy and excess empirical risk guarantees as that of Mechanism PROJERM detailed in Theorem 3.11 with $\psi = O(1)$.

Under $\epsilon$-differential privacy, these are the first bounds that take the geometry of $C$ into account to obtain lower excess.
Mechanism 2 STDVM

**Input:** A dataset \( D = (x_1, y_1), \ldots, (x_n, y_n) \) of \( n \) datapoints from the domain \( Z = \{(x, y) : x \in \mathbb{R}^d, \|x\| \leq 1, y \in \mathbb{R}, |y| \leq 1\} \).

**Output:** \( \theta_{\text{priv}} \) a differentially private estimate of \( \hat{\theta} \in \arg\min_{\theta \in C} \frac{1}{n} \sum_{i=1}^{n} \ell((x_i, \theta); y) \)

1. Generate \( \hat{\Phi} \in \mathbb{R}^{m \times d} \leftarrow \) matrix whose \( m \) rows are i.i.d., mean-zero, isotropic, subgaussian random vectors in \( \mathbb{R}^d \) with \( \psi_2 \)-norm equaling \( O(1) \)
2. \( \Phi \leftarrow \frac{\hat{\Phi}}{\sqrt{m}} \)
3. \( \theta_{\text{priv}} \leftarrow \) Output of Mechanism PROJERM invoked on the input \( (\Phi x_1, y_1), \ldots, (\Phi x_n, y_n) \)
4. **Return:** \( \theta_{\text{priv}} \)

The difference between the compressed setting and the traditional setting is that the \( \psi_2 \)-norm of the matrix used in the compressed setting is bounded by \( O(1) \), while in the traditional setting, the norm could be unbounded.

Efficient Private Empirical Risk Minimization for High-dimensional Learning

Compressed setting: To our best knowledge, these bounds are the first for private compressed learning, which as discussed in the Introduction (Section 1) is a desired framework for designing efficient algorithms for high-dimensional problems. All previous private ERM algorithms, including that of Talwar et al. operate on the original \( d \)-dimensional input space.

**Traditional setting (under \( \epsilon \)-differential privacy):** The bounds of Talwar et al. hold only under the weaker notion of \((\epsilon, \delta)\)-differential privacy (Thakurta, 2016), whereas in this paper we establish bounds under both \( \epsilon \)- and \((\epsilon, \delta)\)-differential privacy.

**Traditional setting (under \((\epsilon, \delta)\)-differential privacy):** Firstly note that the bound of Talwar et al. holds for all Lipschitz convex functions not just generalized linear functions (the focus of this paper). Now translating the results of Talwar et al. to our setting, their mirror descent based algorithm has an expected excess empirical risk bound of \( O(\lambda w(C) \sqrt{\max_{\theta \in C} \Psi(\theta)} \log(n/\delta)/cn) \), where \( C \rightarrow \mathbb{R} \) needs to be picked to be 1-strongly convex with respect to \( \| \cdot \|_C \) norm. Talwar et al. also provide few example instantiations of this algorithm by carefully picking \( \Psi \) based on \( C \). The Frank-Wolfe based algorithm of Talwar et al. has an expected excess empirical risk bound of \( O(\Gamma_\ell^2/3(\lambda w(C))^{2/3} \log^2(n/\delta)/(en)^{2/3}) \), where \( \Gamma_\ell \) is the curvature constant of \( \ell \) over the domain \( C \).

In general, because of the dependence on slightly different parameters, it is hard to precisely compare the various excess risk bounds. Therefore, as an example, consider the problem of linear regression: \( \min_{\theta \in C} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle x_i, \theta \rangle)^2 \). In this case, the expected excess risk bound (using the Frank-Wolfe approach) from Talwar et al. is \( O(||C||_2^4/3 w(C)^{2/3}/(en)^{2/3}) \), whereas the bound from Theorem 3.11 (Part 1) is \( O(||C||_2^4 \sqrt{w(C)}/\sqrt{\gamma n}) \) (for simplification: setting \( \delta \approx 1/\log(n) \)). For bounded \( \|C\|_2, \epsilon \), and in the regime of \( \|w(C)\| = o(n) \), when both our result and that of Talwar et al. are better than the trivial private risk bound of \( O(1) \), the bound of Talwar et al. is better than our bound by a factor of \( \approx n^{1/6}/w(C)^{1/6} \). So even under \((\epsilon, \delta)\)-differential privacy, our bounds are close to that achieved by Talwar et al. This along with previous points underscore the importance of our results.
Acknowledgments

The first author would like to thank his son, Tarun, who thoughtfully prolonged his stay in utero until after his father timely submitted this manuscript.

References


Thakurta, Abhradeep. Personal communication, 2016.


