DOI: 10.1561/9781638284772.ch6

Chapter 6

Private Optimization

By Abhradeep Thakurta

6.1 Introduction

Consider a data set $D = \{d_1, \ldots, d_n\}$ drawn from some domain $\mathcal{D}^*$, and a loss function $\mathcal{L}(\theta; D) = \sum_{i=1}^{n} \ell(\theta; d_i)$, where $\theta \in \mathbb{R}^p$ is the model, and $\ell : \mathbb{R}^p \times \mathcal{D}$ is the loss function on individual data samples. In this chapter, we will focus on algorithms for estimating (6.1) while preserving (ε, δ)-differential privacy, where $\mathcal{C} \subseteq \mathbb{R}^p$ is the constraint set

$$\theta^* \in \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; D). \tag{6.1}$$

The above formulation is often called the Empirical Risk Minization (ERM), and is powerful enough to capture a large class of learning tasks. For example, i) in linear regression the data record is $d_i = (\mathbf{x}_i, y_i)$ (with $\mathbf{x}_i \in \mathbb{R}^d$ being the feature vector, and y_i being the response) and the loss function is $\ell(\theta; d_i) = (y_i - \langle \mathbf{x}_i, \theta \rangle)^2$, ii) in logistic regression the loss function is $\ell(\theta; d_i) = \ln\left(1 + e^{-y_i \langle \mathbf{x}_i, \theta \rangle}\right)$, and iii) in the case of deep networks with binary cross entropy, the loss function is $\ell(\theta; d_i) = \ln\left(1 + e^{-y_i \cdot h_\theta(\mathbf{x}_i)}\right)$, where $h_\theta(\cdot)$ is the network parameterized by the model weights θ. ERM frameworks also allow a direct a direct way of minimizing the population

loss (a.k.a. true risk or the test accuracy), i.e,

$$\theta^*_{\text{pop}} \in \arg\min_{\theta \in \mathcal{C}} \mathbb{E}_{d\sim\mathcal{T}} \left[\ell(\theta; d)\right]. \tag{6.2}$$

In (6.2), $\mathcal{T}$ is a distribution over the domain $\mathcal{D}$. If the data set D is drawn i.i.d. from the distribution $\mathcal{T}$, it is then not hard to show by the so-called uniform convergence theorem [SSSS09] that the following holds for any $\theta \in \mathcal{C}$, with probability at least $1 - \beta$ over the randomness of D. In (6.3), L is the ℓ_2-Lipschitz constant (Defnition 6.1) on the individual loss functions $\ell(\cdot;\cdot)$ w.r.t. the first parameter, and $\|\mathcal{C}\|_2$ is the ℓ_2-diameter of the constraint set $\mathcal{C}$. Hence, if one can estimate θ^* well with differential privacy, then it immediately implies a strong bound on the true risk[i]. In this chapter, we will hence focus on solving the ERM problem (defined in (6.1)) with differential privacy.

$$\begin{aligned}
&\underbrace{\mathbb{E}_{d\sim\mathcal{T}} \left[\ell(\theta; d)\right] - \mathbb{E}_{d\sim\mathcal{T}} \left[\ell(\theta^*_{\text{pop}}; d)\right]}_{\textbf{Excess true risk}} \\
&= \frac{1}{n} \left(\underbrace{\mathcal{L}(\theta; D) - \mathcal{L}(\theta^*; D)}_{\textbf{Excess empirical risk}} \right) \\
&\quad + O\left(L \|\mathcal{C}\|_2 \cdot \sqrt{\frac{p \cdot \ln(n) \cdot \ln(d/\beta)}{n}} \right).
\end{aligned} \tag{6.3}$$

To begin with, we need the following properties of (convex) function that we will be using throughout the chapter. In each of the algorithms, and the corresponding analysis, we will explicitly state which properties of the function we are assuming.

Definition 6.1 (L-Lipschitz continuity)**.** *A function $f : \mathcal{C} \to \mathbb{R}$ is L-Lipschitz w.r.t. the ℓ_q-norm if the following is true for any $\theta_1, \theta_2 \in \mathcal{C}$:*

$$\left|f(\theta_1) - f(\theta_2)\right| \leq L \cdot \|\theta_1 - \theta_2\|_q. \tag{6.4}$$

Unless mentioned explicitly, we will assume Lipschitzness w.r.t. the ℓ_2-norm.

Definition 6.2 (γ -Smoothness)**.** *A function $f : \mathcal{C} \to \mathbb{R}$ is γ -smooth, if the following is true for any $\theta_1, \theta_2 \in \mathcal{C}$:*

$$f(\theta_2) \leq f(\theta_1) + \langle \nabla f(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\gamma}{2} \|\theta_2 - \theta_1\|_2^2. \tag{6.5}$$

i. There are more sophisticated and tighter methods for converting from excess empirical risk to excess true risk [SSSS09; BFTT19], but they are beyond the scope of this chapter.

Definition 6.3 (Δ-Strong convexity)**.** *A function $f : \mathcal{C} \to \mathbb{R}$ is Δ-strongly convex, if the following is true for any $\theta_1, \theta_2 \in \mathcal{C}$ and for any $\alpha \in (0, 1]$:*

$$f(\alpha \cdot \theta_1 + (1-\alpha) \cdot \theta_2) \leq \alpha \cdot f(\theta_1) + (1-\alpha) \cdot f(\theta_2) - \frac{\Delta \cdot \alpha(1-\alpha)}{2} \|\theta_2 - \theta_1\|_2^2 . \tag{6.6}$$

If $\Delta = 0$, we say that the function f is convex. If f is differentiable, we can replace the condition in (6.6) *with,*

$$f(\theta_2) \geq f(\theta_1) + \langle \nabla f(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\Delta}{2} \|\theta_2 - \theta_1\|_2^2 . \tag{6.7}$$

Overview of the Chapter

The chapter is organized as follows. We will first discuss DP-ERM algorithms that satisfy pure ε-differential privacy. The specific algorithms we will discuss are i) Exponential mechanism [MT07; BST14], and ii) Objective perturbation [CMS11; KST12]. Then, we will move on to discuss DP-ERM algorithms that satisfy (ε, δ)-differential privacy. There, we will focus our attention on a couple of algorithms, namely, differentially private (stochastic) gradient descent (DP-SGD) [BST14; Aba+16; TTZ14a] and differentially private follow the regularized leader (DP-FTRL) [Kai+21b; ST13a; AS17]. We will then provide a (ε, δ)-differentially private algorithm for the high-dimensional setting where the dimensionality $p \gg n$. This algorithm is called the private Frank-Wolfe [TTZ15]. Finally, we will demonstrate how (and when) these algorithms provide optimal privacy/utility trade-offs by visiting some of the lower bounding techniques in DP-ERM.

Note: All the results in this section are in the add/remove model of differential privacy (see Section 1.4.1 of Chapter 1). They can be easily translated to the replacement model by paying a factor of two in the privacy parameters via standard methods [DR+14].

6.2 Empirical Risk Minimization with ε-DP

In this section, we provide algorithms for solving the ERM problem in (6.1) under ε-differential privacy.

6.2.1 Exponential Mechanism based Private ERM

The first algorithm we look at is based on the classic exponential mechanism [MT07]. Later we will see that this algorithm is indeed optimal for the case

when for any data sample $d \in \mathcal{D}$, the loss function $\ell(\theta; d)$ is convex and L-Lipschitz in its first parameter, within the convex constraint set $\mathcal{C}$.

Algorithm 1 $\mathcal{A}_{\exp-\text{samp}}$: Exponential mechanism based convex optimization

Require: Data set of size n: D, loss function: ℓ, constraint set: $\mathcal{C}$, ℓ_2-Lipschitz constant: L, privacy parameter: ε

1: $\mathcal{L}(\theta; D) \leftarrow \sum_{i=1}^{n} \ell(\theta; d_i)$.

2: Sample and **output** a point θ^{priv} from the constraint set $\mathcal{C}$ w.p. $\propto \exp\left(-\frac{\varepsilon}{2L\|\mathcal{C}\|_2} \cdot \mathcal{L}(\theta; D)\right)$.

First, we show that Algorithm $\mathcal{A}_{\exp-\text{samp}}$ is ε-differentially private. The proof will go via fairly standard arguments used in analyzing exponential mechanism. A vanilla analysis of the sampling distribution in the algorithm would require the loss $\ell(\theta; \cdot)$ to be bounded in terms of its value. However, by using a fixed anchoring point θ_0, it suffices to operate with the Lipschitzness assumption.

Theorem 6.4. *Algorithm 1 is ε-differentially private.*

Proof. Consider the kernel $\mu(\theta; D) = \exp\left(-\frac{\varepsilon}{2L\|\mathcal{C}\|_2} \cdot \mathcal{L}(\theta; D)\right)$ of the probability distribution in Algorithm $\mathcal{A}_{\exp-\text{samp}}$. Let $\theta_0 \in \mathcal{C}$ be any fixed model parameter. Now, notice that the sampling distribution generated by $\mu(\theta)$ is identical to the distribution generated by the following kernel: $\widehat{\mu}(\theta; D) = \exp\left(-\frac{\varepsilon}{2L\|\mathcal{C}\|_2} \cdot (\mathcal{L}(\theta; D) - \mathcal{L}(\theta_0; D))\right)$. Hence, in the rest of the proof we will only consider $\widehat{\mu}(\theta; D)$.

Consider any two neighboring data sets D and D'. Let d be the data record on which they differ. W.l.o.g., data set D has data record d, and D' does not. For any θ, θ_0, we have the following.

$$\begin{aligned}\left|(\mathcal{L}(\theta; D) - \mathcal{L}(\theta_0; D)) - \left(\mathcal{L}(\theta; D') - \mathcal{L}(\theta_0; D')\right)\right| &= |\ell(\theta; d) - \ell(\theta_0; d)| \\ &\leq L \cdot \|\theta - \theta_0\|_2 \leq L\,\|\mathcal{C}\|_2\,. \end{aligned} \tag{6.8}$$

To ensure differential privacy we need to make sure that for any measurable set $S \subseteq \mathcal{C}$, the following is true.

$$e^{-\varepsilon} \leq \frac{\int_{\theta \in S} \mu\,(\theta; D)}{\int_{\theta \in \mathcal{C}} \mu\,(\theta; D)} \cdot \frac{\int_{\theta \in \mathcal{C}} \mu\,\left(\theta; D'\right)}{\int_{\theta \in S} \mu\,(\theta; D')} \leq e^{\varepsilon}. \tag{6.9}$$

By (6.8), for any $\theta \in \mathcal{C}$, $e^{-\varepsilon/2} \leq \frac{\mu(\theta; D)}{\mu(\theta; D')} \leq e^{-\varepsilon/2}$. Hence, the condition in (6.9) is immediately satisfied. This completes the proof. □

In Theorem 6.5 we show that for Algorithm 1, the excess empirical risk is bounded by $O\left(\frac{pL\|\mathcal{C}\|_2}{\varepsilon}\right)$. In the proof, we will heavily use convexity property of the loss function $\ell(\cdot;\cdot)$ to show this bound.

Note: The following simpler bound is easy to prove without relying on convexity: Let $\mathbb{B}$ be a unit ball centered at the origin. If $r\mathbb{B} \subseteq \mathcal{C}$, then the excess empirical risk is bounded by $O\left(\frac{pL\|\mathcal{C}\|_2}{\varepsilon} \cdot \ln\left(\frac{\varepsilon nL\|\mathcal{C}\|_2}{r}\right)\right)$. We leave the proof of this statement as an exercise. This result is significant because it shows that one can obtain both DP guarantee, and strong excess empirical risk bounds just by assuming the loss function to be L-Lipschitz within the constraint set.

Theorem 6.5. *Let θ^{priv} be the output of Algorithm 1 above. Then, we have the following guarantee on the expected excess risk. (The expectation is over the randomness of the algorithm.)*

$$\mathbf{E}\left[\mathcal{L}(\theta^{priv}; D) - \mathcal{L}(\theta^*; D)\right] = O\left(\frac{pL\,\|\mathcal{C}\|_2}{\varepsilon}\right).$$

Here, $\theta^* \in \arg\min_{\theta\in\mathcal{C}} \mathcal{L}(\theta; D)$.

Proof. Consider any differential cone $\boldsymbol{\Omega}$ centered at θ^*. We will first bound the excess empirical risk, conditioned on $\theta^{\mathsf{priv}} \in \boldsymbol{\Omega}$. Since the bound will be true for any $\boldsymbol{\Omega}$, by the law of total expectation, the guarantee in the theorem statement immediately follows.

Let $\Gamma \geq 0$ be a fixed parameter to be defined later. For the purpose of brevity, let $f(\theta) = \mathcal{L}(\theta; D) - \mathcal{L}(\theta^*; D)$. We first split $\boldsymbol{\Omega}$ into different levels A_i's, where each A_i is defined as follows:

$$A_i = \left\{\theta \in \boldsymbol{\Omega} \cap \mathcal{C} : (i-1)\cdot\Gamma \leq f(\theta) \leq i\cdot\Gamma\right\}. \tag{6.10}$$

Notice that A_1 corresponds to the region, where the excess empirical risk $f(\theta) \leq \Gamma$. Instead of directly computing the probability of θ^{priv} lying outside A_1, we will individually compute the probability of θ^{priv} being in each of $A_i, i > 1$ individually, and then take an union bound over these probabilities. Typically, this line of argument is referred to as the "peeling argument".

Since $\boldsymbol{\Omega}$ is a differential cone, and $f(\theta)$ is continuous on $\mathcal{C}$, it follows that within $\boldsymbol{\Omega} \cap \mathcal{C}$, $f(\theta)$ only depends on $\|\theta - \theta^*\|_2$. Centered at θ^*, let $r_1, r_2, \ldots$ be the end boundaries for the sets $A_1, A_2, \ldots$ respectively. Hence, one can redefine (6.10) as follows.

$$A_i = \left\{\theta \in \boldsymbol{\Omega} \cap \mathcal{C} : r_{i-1} \leq f(\theta) \leq r_i\right\}, \tag{6.11}$$

In Claim 6.6 we show that due to convexity of $f(\theta)$, the gap between successive r_i's (i.e., $r_i - r_{i-1}$) is decreasing for $i \geq 3$.

Claim 6.6. *Convexity of $f(\theta)$ implies that $r_i - r_{i-1} \leq r_{i-1} - r_{i-2}$ for all $i \geq 3$.*

Proof. Since θ^* is the minimizer of $f(\theta)$ within the constraint set $\mathcal{C}$, and since $f(\theta)$ is convex, we have the following any two $\theta_1, \theta_2 \in \mathcal{C} \cap \Omega$ with $f(\theta_2) \geq f(\theta_1)$: $\|\theta_2 - \theta^*\|_2 \geq \|\theta_1 - \theta^*\|_2$. This immediately implies the claim.

Now, recall the volume of any of the A_i is given by $\text{Vol}(A_i) = \text{const} \cdot \int_{r_{i-1}}^{r_i} r^{p-1} dr$. Hence, we have the following:

$$\frac{\text{Vol}(A_i)}{\text{Vol}(A_2)} = \left(\frac{r_{i-1}}{r_1}\right)^p \cdot \frac{(r_i/r_{i-1})^p - 1}{(r_2/r_1)^p - 1} \leq \left(\frac{r_{i-1}}{r_1}\right)^p \leq (i-1)^p. \tag{6.12}$$

□

The last two inequalities in (6.12) follows directly from Claim 6.6. Recall the definition of Γ. Hence we have the following for the excess empirical risk $f(\theta^{\text{priv}}) \geq 4\Gamma$, conditioned on $\theta^{\text{priv}} \in \mathcal{C} \cap \Omega$. In the following, we remove the conditioning for brevity.

$$\mathbf{Pr}[f(\theta^{\text{priv}}) \geq 4\Gamma] \leq \frac{\mathbf{Pr}[\theta^{\text{priv}} \in \bigcup_{i=4}^{\infty} A_i]}{\mathbf{Pr}[\theta^{\text{priv}} \in A_2]} \leq \sum_{i=4}^{\infty} \frac{\text{Vol}(A_i)}{\text{Vol}(A_2)} \cdot \exp\left(-\frac{\varepsilon(i-3)\Gamma}{2L\|\mathcal{C}\|_2}\right) \tag{6.13}$$

$$\leq \sum_{i=4}^{\infty} (i-1)^p \cdot \exp\left(-\frac{\varepsilon(i-3)\Gamma}{2L\|\mathcal{C}\|_2}\right)$$

$$\leq \frac{3^p \exp\left(-\frac{\varepsilon\Gamma}{2L\|\mathcal{C}\|_2}\right)}{1 - 2^p \exp\left(-\frac{\varepsilon\Gamma}{2L\|\mathcal{C}\|_2}\right)}. \tag{6.14}$$

The last inequality in (6.14) follows from the fact that $(i-1)^p \leq 3^p \cdot \left(2^{i-1}\right)^p$ for all $i \geq 4$. Hence for every $t > 0$, if we choose $\Gamma = \frac{2L\|\mathcal{C}\|_2}{\varepsilon} \cdot \big((p+1)\ln(3) + t\big)$, then we have the following.

$$\mathbf{Pr}\left[f(\theta^{\text{priv}}) \geq \frac{8L\|\mathcal{C}\|_2}{\varepsilon} \cdot \big((p+1)\ln(3) + t\big)\right] \leq e^{-t}. \tag{6.15}$$

Since (6.15) is true for all $t \geq 0$, we have the required bound as a corollary. □

Oracle Complexity

It is not obvious how to implement Algorithm 1 efficiently. Even before that we need to decide on how we measure computational complexity. In this section, and in the rest of the chapter, we will measure the complexity in terms of number of

oracle calls to the gradients of individual loss function $\nabla \ell(\theta; d)$. This is consistent with the standard optimization literature [Bub15].

Since the loss functions $\ell(\cdot;\cdot)$ are convex in the first parameter, Step 2 of Algorithm 1 results in sampling from a log-concave distribution. Classic results from sampling theory [LV06] allows sampling efficiently from these distribution, albeit the convergence is usually in the total-variation-distance (TVD). In order to guarantee ε-differential privacy, it is necessary to have the sampling distribution to converge to the true distribution induced by Step 2 of Algorithm 1 in the ℓ_∞-distance. [BST14], which got improved by [MV21], provides algorithms with ℓ_∞-distance convergence and oracle complexity of $O\left(\frac{n}{\varepsilon}\text{poly}\left(p\right)\right)$.

Requirement of Convexity

While the utility analysis of Algorithm 1 heavily relies on convexity, the privacy analysis does not. The privacy analysis only assumes that $\ell(\theta; d)$ is L-Lipschitz w.r.t. the ℓ_2-norm. As it will be more obvious in the later parts of this chapter, such a property is rare in algorithms designed for DP optimization. However, the general philosophy regarding the design of DP optimization algorithms that the reader should keep in mind is that, it is always desirable that the privacy property of an algorithm should rely on minimal set of assumptions, which in particular should be enforceable/efficiently testable. It is okay to make stronger assumptions for the corresponding utility analysis. Convexity, unfortunately, is not an efficiently testable property in general.

Note on Optimality

The utility bound obtained in Theorem 6.5 is indeed optimal. One can use use standard machinery of the so-called "packing argument" [HT10; BST14] to achieve the lower bound.

6.2.2 Objective Perturbation for Private ERM

While Algorithm $\mathcal{A}_{\text{exp}-\text{samp}}$ is optimal for pure ε-DP (i.e., with $\delta = 0$), any natural extension of $\mathcal{A}_{\text{exp}-\text{samp}}$ is not known to be optimal in the case of (ε, δ)-DP. In this section we will see an algorithm that is simultaneously optimal for both ε-DP and (ε, δ)-DP. Additionally, this algorithm will be computationally efficient in regards to oracle complexity. The algorithm is called *objective perturbation* (Algorithm 2) [CMS11; KST12]. Along with the loss function $\ell(\theta;\cdot)$ being L-Lipschitz w.r.t. ℓ_2-norm, it requires two additional properties: i) $\ell(\theta;\cdot)$ should be twice continuously differentiable, ii) $\lambda_{\max}\left(\nabla^2_\theta \ell(\theta;\cdot)\right) \leq \lambda_{\max}$, and iii) $\text{rank}\left(\nabla^2_\theta \ell(\theta;\cdot)\right) \leq r$. Here $\lambda_{\max}(\cdot)$ corresponds to the maximum eigenvalue of a positive semidefinite (PSD) matrix.

Algorithm 2 $\mathcal{A}_{\text{obj}-\text{pert}}$: Objective Perturbation

Require: Data set of size n: D, loss function: ℓ, ℓ_2-Lipschitz constant: L, constraint set: $\mathcal{C}$, ℓ_2-regularization: Δ, noise multiplier: λ.

1: $\mathcal{L}(\theta; D) \leftarrow \sum_{i=1}^{n} \ell(\theta; d_i)$.

2: $\theta^{\text{priv}} \leftarrow \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; D) + \frac{\Delta}{2} \|\theta\|_2^2 + \langle b, \theta \rangle$, where $b \sim \mathcal{G}(L \cdot \lambda)$ and $\mathcal{G}(\sigma) := \frac{\exp\left(-\frac{\|x\|_2}{\sigma}\right)}{\int_{x \in \mathbb{R}^d} \exp\left(-\frac{\|x\|_2}{\sigma}\right)}$.

3: **Output** θ^{priv}.

In Theorem 6.7 we provide the privacy guarantee for Algorithm 2. Notice that the regularization parameter Δ has to be lower bounded to ensure DP. As it will be clear later, this lower bound is much lower than that what one would set to get an optimal excess empirical risk bound. In Theorem 6.7, we only provide the privacy analysis for the setting where the constraint set $\mathcal{C} = \mathbb{R}^p$. While the privacy guarantee holds for any convex constraint set $\mathcal{C} \subseteq \mathbb{R}^p$, the proof is much more involved and requires measure theoretic arguments. We encourage curious readers to look at [KST12] for more details.

The proof of Theorem 6.7 will provide a curious connection to exponential mechanism, which was not known earlier, prior to this book chapter.

Theorem 6.7. *Let the loss function $\ell(\theta; d)$ be twice continuously differentiable for all $\theta \in \mathcal{C}$, and for all $d \in \mathcal{D}$. Furthermore, $\forall \theta \in \mathcal{C}, \forall d \in \mathcal{D} : \|\nabla \ell(\theta; d)\|_2 \leq L$, $\lambda_{\max}\left(\nabla^2 \ell(\theta; d)\right) \leq \lambda_{\max}$ and $\text{rank}\left(\nabla_\theta^2 \ell(\theta; d)\right) \leq r$. If we set the noise multiplier $\lambda = 2/\varepsilon$, and the regularization parameter $\Delta \geq \frac{r\lambda_{\max}}{1-\exp(-\varepsilon/2)}$, then Algorithm 2 (Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$) satisfies ε-differential privacy.*

Proof for the special case when $\mathcal{C} = \mathbb{R}^p$. Consider the regularized loss function $\mathcal{J}(\theta; D) = \mathcal{L}(\theta; D) + \frac{\Delta}{2} \|\theta\|_2^2$. Consider the following probability distribution:

$$\mu_D(\theta) \propto \exp\left(-\frac{\varepsilon}{2L} \cdot \|\nabla \mathcal{J}(\theta; D)\|_2\right) \cdot \frac{1}{\det\left(\nabla^2 \mathcal{J}(\theta; D)\right)}. \tag{6.16}$$

Additionally, let $b_D(\theta) = \nabla \mathcal{J}(\theta; D)$. Since, $\mathcal{J}(\theta; D)$ is a strictly convex function, the mapping is a bijection. Let $\nu_D(b)$ be the induced distribution on the random variable $b_D(\theta)$. By the Radon-Nikodym theorem [Bil08], we have the following:

$$\nu_D(b) \propto \exp\left(-\frac{\varepsilon}{2L} \cdot \|b\|_2\right) \cdot \frac{\det(\nabla^2 \mathcal{J}(\theta; D))}{\det(\nabla^2 \mathcal{J}(\theta; D))} = \exp\left(-\frac{\varepsilon}{2L} \cdot \|b\|_2\right). \tag{6.17}$$

Notice that the induced distribution $\nu_D(b)$ is independent of the data set D. Hence, from here on we will remove the subscript D in the rest of the proof, and refer the

distribution as $\nu(b)$. Going back to (6.16), we want to understand the differential privacy property of $\mu_D(\theta)$. For two neighboring data sets D and D', and at a given θ, we have the following:

$$\frac{\mu_D(\theta)}{\mu_{D'}(\theta)} = \underbrace{\frac{\nu(b_D(\theta))}{\nu(b_{D'}(\theta))}}_{A} \cdot \underbrace{\frac{\det\left(\nabla^2 \mathcal{J}(\theta; D')\right)}{\det\left(\nabla^2 \mathcal{J}(\theta; D)\right)}}_{B}. \tag{6.18}$$

We can easily bound term A in (6.18) by the ℓ_2-Lipschitz property of the loss function $\ell(\cdot;\cdot)$. By definition of $b_D(\theta)$, we have the following:

$$\|b_D(\theta) - b_{D'}(\theta)\|_2 = \left\|\nabla \mathcal{J}(\theta; D) - \nabla \mathcal{J}(\theta; D')\right\|_2 \leq L. \tag{6.19}$$

Therefore, by triangle inequality, the term A in (6.18) is upper bounded by $\exp\left(\frac{\varepsilon}{2}\right)$. Next, we will bound term B in (6.18). For the purpose of brevity, let $W = \nabla^2 \mathcal{J}(\theta; D')$ and $W' = \nabla^2 \mathcal{J}(\theta; D')$. We prove the following claim.

Claim 6.8. *Under the assumptions of Theorem 6.7, we have* $\frac{\det(W')}{\det(W)} \leq \frac{\Delta}{\Delta - r\lambda_{\max}}$.

Proof. Since the rank of any $\nabla^2 \ell(\theta; d)$ is upper bounded by r, it follows that the matrix $E = W' - W$ has rank at most r. Let $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq \Delta$ be the eigenvalues of the matrix W, and $\sigma'_1 \geq \sigma'_2 \geq \ldots \geq \sigma'_p \geq \Delta$ correspondingly for W'. Recall that $\det(W) = \prod_{i=1}^{p} \sigma_i$. Therefore, we have the following.

$$\frac{\det\left(W'\right)}{\det\left(W\right)} = \prod_{i=1}^{p} \frac{\sigma'_i}{\sigma_i} = \prod_{i=1}^{p} \left(1 + \frac{\sigma'_i - \sigma_i}{\sigma_i}\right) \leq \prod_{i=1}^{p} \left(1 + \frac{|\sigma'_i - \sigma_i|}{\Delta}\right) \tag{6.20}$$

$$= 1 + \sum_{i=1}^{p} \frac{|\sigma'_i - \sigma_i|}{\Delta} + \sum_{i,j \in [p], i \neq j} \frac{\prod_{k \in \{i,j\}} |\sigma'_k - \sigma_k|}{\Delta^2} + \cdots \tag{6.21}$$

$$\leq 1 + \frac{r\lambda_{\max}}{\Delta} + \left(\frac{r\lambda_{\max}}{\Delta}\right)^2 + \cdots \leq \frac{\Delta}{\Delta - r\lambda_{\max}}. \tag{6.22}$$

The inequality in (6.20) follows from the fact that each of the eigenvalues are lower bounded by Δ. The first inequality in (6.22) follows from the fact that $\forall i \in [p], |\sigma'_i - \sigma_i| \leq r\lambda_{\max}$. The last inequality in (6.22) follows from fact that $\Delta \geq r\lambda_{\max}$. This completes the proof. □

Since by assumption $\Delta \geq \frac{r\lambda_{\max}}{1 - \exp(-\varepsilon/2)}$, term B in (6.18) is upper bounded by $\exp\left(\frac{\varepsilon}{2}\right)$. Hence, we have proved that the sampling distribution in (6.16) satisfies ε-differential privacy. In the rest of the proof we will show that the distribution of θ^{priv} in Algorithm $\mathcal{A}_{\text{obj-pert}}$ is identical to (6.16).

Since we are operating the in the unconstrained space, i.e., $\mathcal{C} = \mathbb{R}^d$, we have the following:

$$b = -\left(\nabla \mathcal{L}(\theta; D) + \Delta\theta\right) = -\left(\nabla \mathcal{J}(\theta; D)\right). \tag{6.23}$$

By using Radon-Nikodym theorem [Bil08], we have the following distribution on θ^{priv}:

$$\mu(\theta^{\text{priv}}) = \nu(b) \cdot \frac{1}{\det(\nabla^2 \mathcal{J}(\theta; D))}. \tag{6.24}$$

Here, $\nu(b)$ is the is the pdf of the distribution on the random variable b, which is proportional to $\exp\left(-\frac{\varepsilon}{2L} \cdot \|b\|_2\right)$. This completes the proof. □

Analogous to Theorem 6.5, we provide the utility guarantee of Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$ in Theorem 6.9.

Theorem 6.9. *Recall all the parameter choices in Theorem 6.7 for Algorithm 2 (Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$). Additionally, assume $\frac{r\lambda_{\max}}{1-\exp(-\varepsilon/2)} \leq \frac{\sqrt{32}\cdot pL}{\varepsilon\|\mathcal{C}\|_2}$, where* $\text{rank}\left(\nabla^2_\theta \ell(\theta; d)\right) \leq r, \forall \theta \in \mathcal{C}, d \in \mathcal{D}$, *and* $0 \in \mathcal{C}$. *Then, under appropriate choice of the regularization parameter Δ, the following is true:*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}, D) - \mathcal{L}(\theta^*; D)\right] = O\left(\frac{pL\,\|\mathcal{C}\|_2}{\varepsilon}\right).$$

Proof. Consider the regularized loss function $\mathcal{J}(\theta; D) = \mathcal{L}(\theta; D) + \frac{\Delta}{2}\|\theta\|_2^2$, and the noisy regularized loss function $\mathcal{J}_{\text{noisy}}(\theta; D) = \mathcal{J}(\theta; D) + \langle b, \theta\rangle$. Let the following be the minimizers for each of the losses: $\theta^{\text{priv}} = \arg\min_{\theta\in\mathcal{C}} \mathcal{J}_{\text{noisy}}(\theta; D)$, $\widehat{\theta} = \arg\min_{\theta\in\mathcal{C}} \mathcal{J}(\theta; D)$, and $\theta^* = \in \arg\min_{\theta\in\mathcal{C}} \mathcal{L}(\theta; D)$.

By the strong convexity property of $\mathcal{J}_{\text{noisy}}(\theta; D)$, the following is true:

$$\mathcal{J}_{\text{noisy}}(\widehat{\theta}; D) \geq \mathcal{J}_{\text{noisy}}(\theta^{\text{priv}}; D) + \frac{\Delta}{2}\left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.25}$$

$$\Leftrightarrow \mathcal{J}(\widehat{\theta}; D) + \langle b, \widehat{\theta}\rangle \geq \mathcal{J}(\theta^{\text{priv}}; D) + \langle b, \theta^{\text{priv}}\rangle + \frac{\Delta}{2}\left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.26}$$

$$\Leftrightarrow \left(\mathcal{J}(\widehat{\theta}; D) - \mathcal{J}(\theta^{\text{priv}}; D)\right) + \langle b, \widehat{\theta} - \theta^{\text{priv}}\rangle \geq \frac{\Delta}{2}\left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.27}$$

$$\Rightarrow \|b\|_2 \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2 \geq \langle b, \widehat{\theta} - \theta^{\text{priv}} \rangle \geq \frac{\Delta}{2} \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.28}$$

$$\Rightarrow \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2 \leq \frac{2\|b\|_2}{\Delta}. \tag{6.29}$$

In the above (6.28) follows from the fact that $\left(\mathcal{J}(\widehat{\theta}; D) - \mathcal{J}(\theta^{\text{priv}}; D)\right) \leq 0$, and the inequality in (6.29) follows via Cauchy-Schwartz. Using (6.28) we immediately have the following inequality, which bounds the difference $\mathcal{J}(\theta^{\text{priv}}; D) - \mathcal{J}(\widehat{\theta}; D)$.

$$\mathcal{J}_{\text{noisy}}(\widehat{\theta}; D) \geq \mathcal{J}_{\text{noisy}}(\theta^{\text{priv}}; D) + \frac{\Delta}{2} \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.30}$$

$$\Leftrightarrow \mathcal{J}(\widehat{\theta}; D) + \langle b, \widehat{\theta} \rangle \geq \mathcal{J}(\theta^{\text{priv}}; D) + \langle b, \theta^{\text{priv}} \rangle + \frac{\Delta}{2} \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.31}$$

$$\Leftrightarrow \mathcal{J}(\theta^{\text{priv}}; D) - \mathcal{J}(\widehat{\theta}; D) \leq \langle b, \widehat{\theta} - \theta^{\text{priv}} \rangle - \frac{\Delta}{2} \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2^2 \tag{6.32}$$

$$\Rightarrow \mathcal{J}(\theta^{\text{priv}}; D) - \mathcal{J}(\widehat{\theta}; D) \leq \|b\|_2 \cdot \left\|\widehat{\theta} - \theta^{\text{priv}}\right\|_2 \leq \frac{2\|b\|_2^2}{\Delta}. \tag{6.33}$$

Now, we finally bound $\mathcal{L}(\theta^{\text{priv}}; D) - \mathcal{L}(\theta^*; D)$ in terms of $\|b\|_2$. We have the following:

$$\begin{aligned}
\mathcal{L}(\theta^{\text{priv}}; D) - \mathcal{L}(\theta^*; D) &= \left(\mathcal{L}(\theta^{\text{priv}}; D) + \frac{\Delta}{2} \left\|\theta^{\text{priv}}\right\|_2^2\right) \\
&\quad - \left(\mathcal{L}(\theta^*; D) + \frac{\Delta}{2} \|\theta^*\|_2^2\right) \\
&\quad + \frac{\Delta}{2} \left(\|\theta^*\|_2^2 - \left\|\theta^{\text{priv}}\right\|_2^2\right) && (6.34) \\
&\leq \mathcal{J}(\theta^{\text{priv}}; D) - \mathcal{J}(\theta^*; D) + \frac{\Delta}{2} \|\theta^*\|_2^2 && (6.35) \\
&\leq \mathcal{J}(\theta^{\text{priv}}; D) - \mathcal{J}(\widehat{\theta}; D) + \frac{\Delta}{2} \|\theta^*\|_2^2 \\
&\leq \left(\frac{2\|b\|_2^2}{\Delta} + \frac{\Delta}{2} \|\theta^*\|_2^2\right). && (6.36)
\end{aligned}$$

The first inequality in (6.36) follows from the fact that $\mathcal{J}(\theta; D) \geq \mathcal{J}(\widehat{\theta}; D), \forall \theta \in \mathcal{C}$, and the second inequality follows from (6.33). Taking expectation on both sides

of (6.36), and optimizing for $\Delta = \frac{2\cdot\sqrt{\mathbb{E}\left[\|b\|_2^2\right]}}{\|\mathcal{C}\|_2}$, we have the following:

$$\mathbb{E}\left[\mathcal{L}(\theta^{\mathsf{priv}}; D) - \mathcal{L}(\theta^*; D)\right] \leq \frac{2\mathbb{E}\left[\|b\|_2^2\right]}{\Delta} + \frac{\Delta}{2}\left\|\theta^*\right\|_2^2 \leq \frac{2\mathbb{E}\left[\|b\|_2^2\right]}{\Delta} + \frac{\Delta}{2}\|\mathcal{C}\|_2^2$$
$$\leq \|\mathcal{C}\|_2 \cdot \sqrt{\mathbb{E}\left[\|b\|_2^2\right]}. \tag{6.37}$$

Given the distribution on b in Algorithm $\mathcal{A}_{\mathsf{obj-pert}}$, it is not hard to observe that $\|b\|_2 \sim \text{Gamma}\left(p, \frac{\varepsilon}{2L}\right)$. Therefore, by standard properties of Gamma distribution, we have $\mathbb{E}\left[\|b\|_2^2\right] = \frac{4pL^2}{\varepsilon^2} + \frac{4p^2L^2}{\varepsilon^2} \leq \frac{8p^2L^2}{\varepsilon^2}$. Plugging in this bound in (6.37) completes the proof. □

Requirement on Smoothness, and Bounded Rank Hessian, and Convexity

For the privacy analysis in Theorem 6.7, we made these assumptions, beyond just assuming that the loss function $\ell(\cdot;\cdot)$ is ℓ_2-Lipschitz bounded. In the following, we discuss the necessity of these assumptions.

Consider a simple problem where the data sample $d_i \in \mathbb{R}$, and the loss function $\ell(\theta; d_i) = |\theta - d_i|$, i.e., the ERM problem $\arg\min_{\theta\in\mathbb{R}} \sum_{i=1}^n |\theta - d_i|$ is estimating the median of the data set $D = \{d_1, \ldots, d_n\}$, with each d_i being unique. For brevity, assume n is odd, so there is a unique median. Clearly, the function is non-differentiable at $\theta \in \{d_1, \ldots, d_n\}$, and hence non-smooth at those points. We focus on the loss function $\mathcal{L}(\theta; D)$ at $\theta = d_{\text{med}}$. Notice that the slope of $\mathcal{L}(\theta; D)$ in the vicinity of d_{med} is either -1 or $+1$. Now, since $|b|$ exponentially distributed (as $p = 1$), we have $\mathbf{Pr}[|b| \leq 1/2] = 1 - \exp(-\varepsilon/4)$. If the strong convexity term $\Delta = 0$ in Algorithm $\mathcal{A}_{\mathsf{obj-pert}}$, then clearly, with probability at least $1-\exp(-\varepsilon/4)$, Algorithm $\mathcal{A}_{\mathsf{obj-pert}}$ outputs $\theta^{\mathsf{priv}} = d_{\text{med}}$. As d_{med} is a data point in the data set D, it is a direct violation of ε-DP. One might argue that non-zero value of strong convexity parameter Δ will improve the situation. To move the minimizer away from d_{med}, it is necessary that $\Delta \geq \frac{1}{2|d_{\text{med}}|}$ to ensure that the absolute value of the slope of the regularizer at d_{med} is at least $1/2$. Since, d_{med} can be arbitrary close to zero, Δ has to be potentially infinite, which in turn will destroy any utility of the algorithm. This argument shows that smoothness is a necessary condition of Algorithm $\mathcal{A}_{\mathsf{obj-pert}}$ to ensure DP.

While convexity is a necessary condition for the proof of privacy in Algorithm $\mathcal{A}_{\mathsf{obj-pert}}$, it is possible that one may be able to design variants of $\mathcal{A}_{\mathsf{obj-pert}}$ that does not rely on convexity for privacy. Curiously, if one observes the sampling distribution of θ^{priv} in (6.16), then it would be obvious that removing the scaling term with $\det\left(\nabla^2 \mathcal{J}(\theta; D)\right)$ with result in a sampling distribution that can proven to ensure ε-DP without relying on convexity, or even smoothness. (The proof will

be a direct extension of the privacy theorem for Algorithm $\mathcal{A}_{\text{exp}-\text{samp}}$, i.e., Theorem 6.4.) It is an active area of research to design variants of Objective perturbation to be amenable to non-convex losses [NRVW20]. There is not much of an intuition there whether the condition on the regularization parameter Δ should necessarily depend on the rank of the Hessian (r). It may be a slack in the analysis of Theorem 6.7.

Computational Efficiency

While Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$ is a mathematically well-defined object, it is unclear how to implement it in practice. In particular, for arbitrary convex losses, any reasonable optimization procedure will not reach the true minimizer θ^{priv} with finite oracle complexity. Given a pre-specified parameter γ, there are optimization methods [Bub15] that will be ensure that they will output a model $\theta^{\dagger}$ s.t. $\left\|\nabla \mathcal{J}_{\text{noisy}}(\theta^{\dagger}; D)\right\|_2 \leq \gamma$ with $O\left(n/\text{poly}(\gamma)\right)$ oracle complexity. (Here $\mathcal{J}_{\text{noisy}}(\theta; D) = \mathcal{L}(\theta; D) + \frac{\Delta}{2}\|\theta\|_2^2 + \langle b, \theta\rangle$.) In the unconstrained setting $\mathcal{C} = \mathbb{R}^p$, because of Δ-strong convexity of $\mathcal{J}_{\text{noisy}}$, it implies $\left\|\theta^{\dagger} - \theta^{\text{priv}}\right\|_2 = O\left(\frac{\gamma}{\Delta}\right)$. Therefore, one can add an additional DP-friendly noise with standard deviation $O\left(\frac{\gamma}{\Delta\varepsilon}\right)$, to cover for $\left\|\theta^{\dagger} - \theta^{\text{priv}}\right\|_2$. For the constrained setting, one can first perform the above procedure on the unconstrained problem, and then project onto the constraint set $\mathcal{C}$. For a detailed discussion on this approach, see [Iye+19; BFTT19].

Excess Empirical Risk for Strongly Convex Functions

Theorem 6.9 is stated for just Lipschitz convex functions. However, the proof essentially goes via bounding the excess empirical risk for the strongly convex objective $\mathcal{J}(\theta; D) = \mathcal{L}(\theta; D) + \frac{\Delta}{2}\|\theta\|_2^2$ (see (6.33)). Using this machinery, one can obtain an excess empirical risk of $O\left(\frac{L^2 p^2}{\Delta n \varepsilon^2}\right)$, if each of the individual loss function $\ell(\theta; d)$ is assumed to be Δ-strongly convex. Notice that this bound is tight [BST14]. While Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$ requires assumptions like smoothness for the privacy proof, a variant of the exponential mechanism $\mathcal{A}_{\text{exp}-\text{samp}}$ can also achieve a similar bound, albeit a more complicated analysis. (See Section 4 of [BST14] for more details.)

Extension to (ε, δ)-DP Variant

We will discuss algorithms that are specifically designed to obtain strong privacy/utility trade-offs in the (ε, δ)-DP setting. But it is worth mentioning that Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$ can be shown to provide strong privacy/utility trade-offs in the (ε, δ)-DP setting too. The only change that is needed is the following: Change the noise distribution of b to $\mathcal{N}\left(0, O\left(\frac{L\sqrt{\ln(1/\delta)}}{\varepsilon}\right)^2 \cdot \mathbb{I}_p\right)$. Since Gaussian distribution has a tighter concentration, the excess empirical risk becomes

$O\left(\frac{L\|\mathcal{C}\|_2 \cdot \sqrt{p \ln(1/\delta)}}{\varepsilon}\right)$. This bound is also known to be tight [BST14]. In Section 6.3 we will study algorithms that achieve similar bounds, however, do not require assumptions like smoothness, or bounded rank of the Hessian. Also, the privacy guarantee of these algorithms will not depend on the convexity of the loss function.

6.3 Empirical Risk Minimization with (ε, δ)-DP

In this section we will look at algorithms that achieve optimal privacy/utility trade-offs under (ε, δ)-DP, while assuming the loss function $\ell(\theta; d)$ being L-Lipschitz in θ, w.r.t. ℓ_2-norm. As we discussed earlier $\mathcal{A}_{\text{obj-pert}}$ achieves similar bounds, but require additional assumptions like smoothness, and bounded rank of the Hessian. Algorithmically, the main difference from Algorithms 1 or Algorithm 2 is that we will not argue privacy for the final model θ^{priv}. Rather, the privacy guarantee will hold for the complete optimization path (i.e., the intermediate models that will get generated. This eventually will imply the (ε, δ)-DP guarantee of the final model θ^{priv}. Since we "privatize" the complete optimization path, as opposed to arguing privacy for the final model θ^{priv}, the resulting algorithms operate under weaker privacy assumptions.

6.3.1 Differentially Private (Stochastic) Gradient Descent (DP-SGD)

DP-SGD [SCS13; BST14; Aba+16] is currently the most widely-used differentially private learning algorithm in practice, with at least two large-scale open source implementations TensorFlow-Privacy [Aba+15], and Opacus [You+21]. Along with its practical success, it also provides optimal privacy/utility trade-offs analytically. While there are various variants of DP-SGD in the literature [SSTT21], the one we will primarily focus on in this chapter is the full-gradient descent version. The presentation of the algorithm (Algorithm 3) will be primarily based on [SSTT21].

There are a few distinctive properties of the algorithm. First, notice the term *clipping norm*. Unlike Algorithms $\mathcal{A}_{\text{exp-samp}}$ and $\mathcal{A}_{\text{obj-pert}}$ above, Algorithm $\mathcal{A}_{\text{DP-SGD}}$ does not assume that the loss function $\ell(\theta; d)$ is ℓ_2-Lipschitz. Rather via the clipping norm bound, in Step 3, it enforces that the ℓ_2-norm of the gradient[ii] of any individual $\ell(\theta; d_i)$ is bounded by L. Second, as we mentioned earlier,

ii. All the results for Algorithm $\mathcal{A}_{\text{DP-SGD}}$ holds if the gradient is replaced by subgradient. So, differentiability is not a necessary condition.

Algorithm 3 $\mathcal{A}_{\text{DP-SGD}}$: Differentially private stochastic gradient descent (DP-SGD)

Require: Data set $D = \{d_1, \cdots, d_n\}$, loss function: $\ell : \mathbf{R}^p \times \mathcal{D} \to \mathbf{R}$, clipping norm: L, constraint set: $\mathcal{C} \subseteq \mathbf{R}^p$, number of iterations: T, noise multiplier: λ, learning rate: η.

1: $\theta_0 \leftarrow 0$.
2: **for** $t = 0, \ldots, T-1$ **do**
3: $g_t = \sum_{i=1}^{n} \text{clip}\left(\nabla \ell(\theta_t; d_i)\right)$, where $\text{clip}(v) = v \cdot \min\left\{1, \frac{L}{\|v\|_2}\right\}$.
4: $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}\left(\theta_t - \eta\left(g_t + \mathcal{N}\left(0, \sigma^2\right)\right)\right)$, where $\Pi_{\mathcal{C}}(\text{v}) = \arg\min_{\theta \in \mathcal{C}} \|v - \theta\|_2$ and $\sigma = L \cdot \lambda$.
5: **end for**
6: **return** $\theta^{\text{priv}} = \frac{1}{T}\sum_{t=1}^{T} \theta_t$.

we will show that the entire optimization path $\{\theta_0, \ldots, \theta_T\}$ is (ε, δ)-DP. Third, in Step 6 we output the average of all the models. One can provide similar utility/privacy trade-off for the last iterate too, i.e., θ_T [BST14]. We chose the average model purely for the simplicity of analysis. Fourth, unlike traditional SGD (stochastic gradient descent) [Bub15], where each g_t is computed on a minibatch of examples from the training set D, in Algorithm $\mathcal{A}_{\text{DP-SGD}}$ we use the complete gradient. As we will discuss later, the privacy/utility trade-off that we will obtain for Algorithm $\mathcal{A}_{\text{DP-SGD}}$ can also be obtained via using a minibatch of size one in each step $t \in [T]$, drawn i.i.d. from D. For the purpose of brevity, we will provide the main analysis in the context of full-batch gradient.

In the following, we first provide the privacy guarantee for Algorithm $\mathcal{A}_{\text{DP-SGD}}$.

Theorem 6.10. *If we choose the noise multiplier* $\lambda = \frac{\sqrt{2T(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$, *then Algorithm* $\mathcal{A}_{\text{DP-SGD}}$ *(Algorithm 3) satisfies* (ε, δ)*-differential privacy.*

Proof. Consider the estimation of any g_t in Step 3 of Algorithm $\mathcal{A}_{\text{DP-SGD}}$, given θ_t. Let D and D' be two neighboring data sets, and g_t and g'_t be the corresponding gradient estimates at θ_t. Due to the clip$(\cdot)$ function, we have the following: $\left\|g_t - g'_t\right\|_2 \leq L$. Therefore, we can think $g_t : t \in [T]$ to be a set of T adaptively chosen queries on the data set D, with each query having ℓ_2-sensitivity of L. By standard composition property of Gaussian mechanism (described in Part I of the book), if we choose the noise scale to be $\lambda = \frac{\sqrt{2T(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$, then the set of $\{\theta_1, \ldots, \theta_t\}$ satisfies (ε, δ)-differential privacy. □

Next, we move on to prove the utility guarantee of Algorithm $\mathcal{A}_{\text{DP-SGD}}$. We show the following:

Theorem 6.11. *Assume that* $\|\nabla \ell(\theta; d)\|_2 \leq L$ *for all* $d \in \mathcal{D}$ *and* $\theta \in \mathcal{C}$, *and* $0 \in \mathcal{C}$ *Under appropriate choice of the learning rate* η, *and setting the number of steps* $T = \frac{n^2\varepsilon^2}{p}$ *and the noise multiplier* $\lambda = \frac{\sqrt{2T(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$, *we have the following:*

$$\mathbb{E}\left[\mathcal{L}(\theta^{priv}, D) - \mathcal{L}(\theta^*; D)\right] = O\left(\frac{L\|\mathcal{C}\|_2}{\varepsilon}\sqrt{p(\ln(1/\delta)+\varepsilon)}\right).$$

Proof. We prove the theorem via the standard template for analyzing SGD methods [Bub15]. Recall $\theta^{\text{priv}} = \frac{1}{T}\sum_{t=1}^{T}\theta_t$, where $\{\theta_1, \ldots, \theta_T\}$ are the models in each iterate of DP-GD. By convexity, and the standard linearization trick in convex optimization [Bub15], we have:

$$\mathcal{L}\left(\theta^{\text{priv}}; D\right) - \mathcal{L}\left(\theta^*; D\right) \leq \frac{1}{T}\sum_{t=1}^{T}\langle \nabla \mathcal{L}(\theta_t; D), \theta_t - \theta^*\rangle. \tag{6.38}$$

Let b_t is the Gaussian noise vector added at time step t To bound the error in (6.38), we will use a potential argument w.r.t. the potential function

$$\Psi_t(\theta) = \mathbb{E}_{b_1,\ldots,b_t}\left[\left\|\theta - \theta^*\right\|_2^2\right] = \mathbb{E}_{b_1,\ldots,b_{t-1}}\left[\mathbb{E}_{b_t}\left[\left\|\theta - \theta^*\right\|_2^2 \middle| b_1, \ldots, b_{t-1}\right]\right].$$

Recall the update step in Algorithm $\mathcal{A}_{\text{DP-SGD}}$: $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}(\theta_t - \eta(\nabla \mathcal{L}(\theta_t; D) + b_t))$. Since, by assumption the loss function $\ell(\theta; d)$ is ℓ_2-Lipschitz bounded, clip$(\cdot)$ does not have any effect. We get the following by simple algebraic manipulation:

$$\begin{aligned}
\Psi_t(\theta_{t+1}) &= \mathbb{E}_{b_1,\ldots,b_t}\left[\left\|\Pi_{\mathcal{C}}\left(\theta_t - \eta(\nabla \mathcal{L}(\theta_t; D) + b_t)\right) - \theta^*\right\|_2^2\right] && (6.39)\\
&\leq \mathbb{E}_{b_1,\ldots,b_t}\left[\left\|\left(\theta_t - \theta^*\right) - \eta(\nabla \mathcal{L}(\theta_t; D) + b_t)\right\|_2^2\right] && (6.40)\\
&= \Psi_t(\theta_t) - 2\eta\mathbb{E}_{b_1,\ldots,b_t}\left[\langle \nabla \mathcal{L}(\theta_t; D) + b_t, \theta_t - \theta^*\rangle\right]\\
&\quad + \eta^2\mathbb{E}_{b_1,\ldots,b_t}\left[\|\nabla \mathcal{L}(\theta_t; D) + b_t\|_2^2\right] && (6.41)\\
&\leq \Psi_t(\theta_t) - 2\eta\mathbb{E}_{b_1,\ldots,b_t}\left[\langle \nabla \mathcal{L}(\theta_t; D), \theta_t - \theta^*\rangle\right]\\
&\quad + \eta^2\left(n^2L^2 + \mathbb{E}_{b_t}\left[\|b_t\|_2^2\right]\right) && (6.42)\\
&= \Psi_{t-1}(\theta_t) - 2\eta\mathbb{E}_{b_1,\ldots,b_t}\left[\langle \nabla \mathcal{L}(\theta_t; D), \theta_t - \theta^*\rangle\right]\\
&\quad + \eta^2L^2\left(n^2 + p\cdot\lambda^2\right), && (6.43)
\end{aligned}$$

where (6.40) follows from the fact that ℓ_2-projection onto a convex constraint set $\mathcal{C}$ always reduces the ℓ_2-distance, and (6.43) follows because $b_t \sim \mathcal{N}(0, L^2\lambda^2)^p$ and thus $\mathbb{E}_{b_t}\left[\|b_t\|_2^2\right] = p\cdot L^2\lambda^2$. Rearranging the terms in (6.43), we have the

following,

$$\mathbb{E}\left[\langle \nabla \mathcal{L}(\theta_t; D), \theta_t - \theta^* \rangle\right] \leq \frac{1}{2\eta}\left(\Psi_{t-1}(\theta_t) - \Psi_t(\theta_{t+1})\right) + \frac{\eta L^2}{2}\left(n^2 + p \cdot \lambda^2\right). \tag{6.44}$$

Summing up (6.44) for all $t \in [T]$, averaging over the T iterations, and combining with (6.38), we get:

$$\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}; D\right)\right] - \mathcal{L}\left(\theta^*; D\right) \leq \frac{1}{2T\eta}\Psi(0) + \frac{\eta L^2}{2}\left(n^2 + p \cdot \lambda^2\right), \quad \text{where } \Psi(\theta) = \left\|\theta - \theta^*\right\|_2^2 \tag{6.45}$$

Setting η to minimize the RHS, we have

$$\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}; D\right)\right] - \mathcal{L}\left(\theta^*; D\right) \leq L\left\|\mathcal{C}\right\|_2 \cdot \sqrt{\frac{n^2 + p \cdot \lambda^2}{T}} \tag{6.46}$$

$$= L\left\|\mathcal{C}\right\|_2 \sqrt{\frac{n^2}{T} + \frac{2p \cdot (\ln(1/\delta) + \varepsilon)}{\varepsilon^2}}, \tag{6.47}$$

where the equality in (6.47) follows by plugging in $\lambda = \frac{\sqrt{2T(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$. Now, setting $T = \frac{n^2\varepsilon^2}{p}$, we have

$$\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}; D\right)\right] - \mathcal{L}\left(\theta^*; D\right) \leq \frac{L\left\|\mathcal{C}\right\|_2 \sqrt{3p \cdot (\ln(1/\delta) + \varepsilon)}}{\varepsilon}. \tag{6.48}$$

This completes the proof. □

Oracle Complexity

Notice, Algorithm $\mathcal{A}_{\mathrm{DP-SGD}}$ reaches an *average* excess empirical risk of $\widetilde{O}\left(\frac{\sqrt{p}}{\varepsilon n}\right)$ in $T = \frac{n^2\varepsilon^2}{p}$ steps. For non-smooth, and non-strongly convex losses, this rate of convergence is tight for SGD based methods (up to dependence on dimensionality), i.e., in $\Theta(1/\alpha^2)$ steps, one can get to an error of α. This demonstrates an important phenomenon. DP does not slow down the rate of convergence in comparison to a non-private SGD method up to the error allowed under privacy constraints. Furthermore, even if we set $T \to \infty$, under appropriate choice of the learning rate η, the excess empirical risk remains the same. This implies, Algorithm $\mathcal{A}_{\mathrm{DP-SGD}}$ does not need the number of steps T to tuned for optimal privacy/utility trade-off as long as $T \geq \frac{n^2\varepsilon^2}{p}$. This property is not true in general for iterative DP optimization methods.

The oracle complexity of Algorithm $\mathcal{A}_{\text{DP-SGD}}$ (for achieving the excess empirical risk in Theorem 6.11) is $\frac{n^3\varepsilon^2}{p}$. This because in each of the T steps, Algorithm $\mathcal{A}_{\text{DP-SGD}}$ performs n gradient evaluations.

Improving Oracle Complexity with Privacy Amplification by Sampling

One can improve the oracle complexity of Algorithm $\mathcal{A}_{\text{DP-SGD}}$ via special tool in the DP literature called, privacy amplification by sampling [Kas+08]. Informally, privacy amplification by sampling says that if an algorithm $\mathcal{A}$ is $\varepsilon_0 \leq \frac{1}{2}$-DP on a data set D, then on a data set D_{samp}, where each entry of D is sampled with probability q, $\mathcal{A}(D_{\text{samp}})$ satisfies $O(q \cdot \varepsilon_0)$-DP. One can use this tool to show that essentially at the same level of noise as in Step 4 of Algorithm $\mathcal{A}_{\text{DP-SGD}}$, one can use $g_t = n \cdot \text{clip}\left(\nabla \ell(\theta_t; d)\right)$ (in Step 3) with d sampled uniformly at random from the data D, independently at each time step T. Since, modulo clipping, the new g_t is an unbiased estimator of the gradient $\nabla \mathcal{L}(\theta_t; D)$, the utility guarantee in Theorem 6.11 remain unchanged. This reduces the oracle complexity by a factor of n for Algorithm $\mathcal{A}_{\text{DP-SGD}}$. For a focused analysis of privacy amplification via sampling, the reader is referred to Chapter 3.6.

Excess ERM Bound for Strongly Convex Losses

In Theorem 6.11 we provided the guarantee only for ℓ_2-Lipschitz convex losses. One can use the same Algorithm $\mathcal{A}_{\text{DP-SGD}}$, with appropriate learning rate η, to obtain optimal privacy/utility trade-off when the loss function $\ell(\theta; d)$ satisfies Δ-strong convexity, along with L-Lipschitzness. The excess empirical risk bound in that case is $O\left(\frac{Lp \ln^3(n/\delta)}{\Delta n \varepsilon^2}\right)$ [BST14]. As we will discuss later, this bound is tight.

Dimension Independent Excess ERM Bounds

In all the results we saw so far there is an explicit polynomial dependence of the dimensionality
(p) on the error. For of generalized linear models (e.g., logistic regression), one can completely avoid this dependence when $\mathcal{C} = R^p$ and achieve an excess empirical risk of $O\left(\frac{L\|\theta^*\|_2\sqrt{n \ln(1/\delta)}}{\varepsilon}\right)$. Algorithm $\mathcal{A}_{\text{DP-SGD}}$, with appropriate choice of the learning rate η is capable of achieving this bound. (See [STT20] for more details.)

6.3.2 Differentially Private Follow-the-regularized-leader (DP-FTRL)

Till now we assumed that the complete data set D is at the disposal of the optimization algorithm. However, there are problem settings where the data arrives in

the form of a stream. Non-privately, the algorithms that are designed in that space are typically called online convex optimization (OCO) algorithms [Sha+11]. More formally, suppose we have a stream of data samples $D = [d_1, \ldots, d_n] \in \mathcal{D}^n$, where $\mathcal{D}$ is the domain of data samples, and a loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$, where $\mathcal{C} \in \mathbb{R}^p$ is the space of all models. We consider the setting of regret minimization.

Regret Minimization

At every time step $t \in [n]$, while observing samples $\{d_1, \ldots, d_{t-1}\}$, the algorithm $\mathcal{A}$ outputs a model $\theta_t \in \mathcal{C}$ which is used to predict on example d_t provided by an adversary after observing $\{\theta_1, \ldots, \theta_t\}$. The performance of $\mathcal{A}$ is measured in terms of regret against an arbitrary post-hoc comparator $\theta^* \in \mathcal{C}$, maximized over the choice of $[d_1, \ldots, d_n]$ by the adversary, where d_t is a function of $\{d_1, \ldots, d_{t-1}\}$ and $\{\theta_1, \ldots, \theta_t\}$:

$$R_D(\mathcal{A}; \theta^*) = \left[\sum_{t=1}^{n} \ell(\theta_t; d_t) - \sum_{t=1}^{n} \ell(\theta^*; d_t) \right]. \tag{6.49}$$

We consider the algorithm $\mathcal{A}$ low-regret if $R(\mathcal{A}; \theta^*) = o(n)$. To ensure a low-regret algorithm, we will assume $\|\nabla \ell(\theta; d)\|_2 \leq L$ for any data sample d, and any models $\theta \in \mathcal{C}$. The quantity $R_D(\mathcal{A}; \theta^*)$ is also called the *adversarial regret* [Haz19; Sha+11]. One can also look at a related quantity *stochastic regret* [HK14], where the data samples in D are drawn i.i.d. from some distribution τ. In this chapter, we will only focus on adversarial regret.

In (6.54), we show that the regret is an upper bound on the excess empirical risk in (6.3). Hence, for the remainder of this section, we will only focus on bounding $R(\mathcal{A}; \theta^*)$. Although we have not discussed the DP-FTRL algorithm ($\mathcal{A}_{\text{FTRL}}$), we want to highlight a specific attribute of the algorithm $\mathcal{A}_{\text{FTRL}}$ in order to connect the regret to the excess empirical risk. At any time step t, to estimate θ_{t+1}, $\mathcal{A}_{\text{FTRL}}$ only uses DP estimates (via Gaussian mechanism) for a set of queries of the form $\sum_{i=\tau_1}^{\tau_2} \nabla \ell(\theta_i; d_i)$, where $0 \leq \tau_1, \tau_2 \leq t$.

Now, given the data set D, consider another data set $\widehat{D}$ with n data samples with each entry of $\widehat{D} = \{\widehat{d}_1, \ldots, \widehat{d}_n\}$ is sampled i.i.d. with replacement from D. First notice that with probability at least $1 - \delta$, no data sample in D appears more that $O(\ln(n/\delta))$ number of times. This follows from the standard use of Chernoff bound. Because of the way $\mathcal{A}_{\text{FTRL}}$ operates, this would increase the ℓ_2-sensitivity of any query that $\mathcal{A}_{\text{FTRL}}$ considers from L to $L \cdot \ln(n/\delta)$, with probability $1 - \delta$. This in-turn means if adding $\mathcal{L}(0, L^2\lambda^2)$ to each query that $\mathcal{A}_{\text{FTRL}}(D)$ considers satisfies (ε, δ)-DP, then adding $\mathcal{N}(0, O\left(L^2 \ln^2(n/\delta)\lambda^2\right))$ satisfies (ε, δ)-DP for $\mathcal{A}_{\text{FTRL}}$ when operating on data set $\widehat{D}$. This calculation says that if $\mathcal{A}_{\text{FTRL}}$ operates on $\widehat{D}$ instead of D, then the error due to noise will only go up by a factor of polylog (n/δ).

Next we relate $\mathcal{L}(\theta; D)$ with $\mathcal{L}(\theta; \widehat{D})$, to ensure that one can use $\widehat{D}$ as a proxy for the data set D. Let $\{\theta_1, \ldots, \theta_n\}$ be the models output by $\mathcal{A}_{\mathrm{FTRL}}$ on data set $\widehat{D}$. We have the following in (6.54).

$$\mathcal{L}\left(\frac{1}{n}\sum_{t=1}^{n}\theta_t; D\right) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) \leq \frac{1}{n}\left(\sum_{t=1}^{n}\mathcal{L}\left(\theta_t; D\right)\right) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) \quad (6.50)$$

$$= \sum_{t=1}^{n}\left(\frac{1}{n}\sum_{i=1}^{n}\ell(\theta_t; d_i)\right) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) \quad (6.51)$$

$$= \sum_{t=1}^{n}\mathbb{E}_{\widehat{d}_t}\left[\ell(\theta_t; \widehat{d}_t)\right] - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) \quad (6.52)$$

$$= \mathbb{E}_{\widehat{D}}\left[\sum_{t=1}^{n}\ell(\theta_t; \widehat{d}_t)\right] - \min_{\theta\in\mathcal{C}}\mathbb{E}_{\widehat{D}}\left[\mathcal{L}(\theta; \widehat{D})\right] \quad (6.53)$$

$$\leq \mathbb{E}_{\widehat{D}}\left[\sum_{t=1}^{n}\ell(\theta_t; \widehat{d}_t) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; \widehat{D})\right] \quad (6.54)$$

$$\leq \mathbb{E}_{\widehat{D}}\left[R_D\left(\mathcal{A}_{\mathrm{FTRL}}; \arg\min_{\theta\in\mathcal{C}}\mathcal{L}\left(\theta; \widehat{D}\right)\right)\right]. \quad (6.55)$$

Equation (6.50) follows from Jensen's inequality, and the equality in (6.52) follows from the fact that θ_t is independent of $\widehat{d}_t$. By (6.54) it follows that the regret of $\mathcal{A}_{\mathrm{FTRL}}$ on data set $\widehat{D}$ is an upper bound on the excess empirical risk of $\theta^{\mathsf{priv}} = \frac{1}{n}\sum_{t=1}^{n}\theta_t$ on the data set D.

Notion of Privacy

Since we are in the add/remove model of differential privacy, in the streaming setting removing one data sample can change time of arrival of all other data samples appearing after it. To ensure that such a thing does not happen, we operate with a notion of differential privacy, that preserves the length of the stream, even in the add/remove variant.

Definition 6.12 (Differential privacy)**.** *Let $\mathcal{D}$ be the domain of data records, $\perp \notin \mathcal{D}$ be a special element, and let $\widehat{\mathcal{D}} = \mathcal{D} \cup \{\perp\}$ be the extended domain. A randomized algorithm $\mathcal{A} : \widehat{\mathcal{D}}^n \to \mathcal{S}$ is (ε, δ)-differentially private if for any data set $D \in \widehat{\mathcal{D}}^n$ and any neighbor $D' \in \widehat{\mathcal{D}}^n$ (formed from D by replacing one record with $\perp$), and for any*

event $S \in \mathcal{S}$, we have

$$\mathbf{Pr}[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{A}(D') \in S] + \delta, \quad \textit{and}$$
$$\mathbf{Pr}[\mathcal{A}(D') \in S] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{A}(D) \in S] + \delta,$$

where the probability is over the randomness of $\mathcal{A}$.

In Algorithm $\mathcal{A}_{\text{FTRL}}$ (Differentially Private Follow-the-regularized-leader (DP-FTRL)), we treat $\perp$ specially, namely assuming it always produces a zero gradient.

Interlude on Estimating Prefix-sum with (ε, δ)-DP

Before we present the description of Algorithm $\mathcal{A}_{\text{FTRL}}$, we will take an interlude to a seemingly unrelated problem of estimating prefix sums: Consider a sequence of vectors $X = \{x_1, \ldots, x_n\}$, with each $x_i \in \mathbb{R}^p$ and $\|x_i\|_2 \leq L$.

The objective it design a differentially private algorithm that outputs an approximation to $\{s_1, \ldots, s_t\}$, where each $s_t = \sum_{i=1}^{t} x_i$. We allow x_{t+1} to be *adaptively* chosen based on the outputs $\{s_1, \ldots, s_t\}$. [DNPR10; CSS11] studies this problem in the context of *privacy under continual observation*. Here, we provide the algorithm from [DNPR10] based on a binary tree data structure.

A naïve solution to the problem would be the following: $\forall t \in [n]$, output $s_t^{\text{priv}} \leftarrow s_t + \mathcal{N}\left(0, L^2\lambda^2\right)$, where $\lambda = \frac{\sqrt{2n(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$. By standard properties of Gaussian mechanism described earlier in the book, this algorithm is (ε, δ)-differentially private. However, the error in each of the estimate s_t^{priv}, i.e., $\mathbb{E}\left[\left\|s_t - s_t^{\text{priv}}\right\|_2\right]$, is $\Omega\left(\frac{\sqrt{np\ln(1/\delta)}}{\varepsilon}\right)$. Based on an algorithm by [DNPR10; CSS11], we provide an algorithm that reduces the error to $\Omega\left(\frac{\sqrt{n\ln^2(n)\cdot\ln(1/\delta)}}{\varepsilon}\right)$. In the following we describe the algorithm.

There are three main functions in the algorithm, namely, InitializeTree, AddToTree, GetSum. At a high-level, InitializeTree initializes the tree data structure $\mathcal{T}$ with $2^{\lceil \lg(n) \rceil}$ leaf nodes, AddToTree allows adding a new vector x_t to $\mathcal{T}$, and GetSum returns the prefix sum $\sum_{i=1}^{t} x_i$ privately. It follows from [ST13a] that for a sequence of (adaptively chosen) vectors $\{x_t\}_{t=1}^{n}$, if we perform AddToTree $(\mathcal{T}, t, x_t)$ for each $t \in [n]$, then we can write GetSum $(\mathcal{T}, t) = \sum_{i=1}^{t} x_i + b_t$ where b_t is normally distributed with mean zero, and $\forall t \in [n], \mathbb{E}[\|b_t\|_2] \leq L\lambda\sqrt{p\lceil \lg(n) \rceil}$. Now, since any data sample in the data set X affects only $\lceil \lg(n) \rceil$ nodes in the binary tree $\mathcal{T}$, hence setting $\lambda = \frac{\sqrt{2\lceil \lg(n) \rceil(\ln(1/\delta)+\varepsilon)}}{\varepsilon}$ would ensure that the algorithm is (ε, δ)-differentially private. The formal description of the algorithm is given below.

1. InitializeTree (n, λ^2, L): Initialize a complete binary tree $\mathcal{T}$ with $2^{\lceil \lg(n) \rceil}$ leaf nodes, with each node being sampled i.i.d. from $\mathcal{N}(0, L^2\lambda^2 \cdot \mathbb{I}_{p\times p})$.

2. AddToTree $(\mathcal{T}, t, v)$: Add v to all the nodes along the path to the root of $\mathcal{T}$, starting from t-th leaf node.
3. GetSum $(\mathcal{T}, t)$: Let $[\text{node}_1, \ldots, \text{node}_h]$ be the list of nodes from the root of $\mathcal{T}$ to the t-th leaf node, with node_1 being the root node and node_h being the leaf node.
   (a) Initialize $s \leftarrow \mathbf{0}^p$ and convert t to binary in h bit representation $[b_1, \ldots, b_h]$, with b_1 being the most significant bit.
   (b) For each $j \in [h]$, if $b_j = 1$, then add the value in left sibling of node_j to s. Here if node_j is the left child, then it is treated as its own left sibling.
   (c) Return s.

We have the following theorem to formally quantify the privacy/utility trade-off for the tree aggregation algorithm.

Theorem 6.13 (Follows from [ST13a]). *Let $X = \{x_1, \ldots, x_n\}$ be a sequence of adaptively chosen data vectors with $\forall u \in [n], \|x_i\|_2 \leq L, x_i \in \mathbb{R}^p$. The tree aggregation algorithm described above is (ε, δ)-differentially private with noise multiplier $\lambda = \frac{\sqrt{2\lceil \lg(n) \rceil (\ln(1/\delta)+\varepsilon)}}{\varepsilon}$. Furthermore, the outputs s_t^{priv} that approximates $s_t = \sum_{i=1}^{t} x_t$ has the following property:*

$$\mathbb{E}\left[\left\| s_t^{\text{priv}} - s_t \right\|_2\right] \leq L\lambda\sqrt{p\lceil \lg(n) \rceil}.$$

Algorithmic Description of DP-FTRL ($\mathcal{A}_{\text{FTRL}}$)

The main idea of DP-FTRL [Kai+21b; ST13a; AS17] is based on three observations: i) For online convex optimization, to bound the regret, for a given loss function $\ell(\theta; d_t)$ (i.e., the loss at time step t), it suffices for the algorithm to operate on a linearization of the loss at θ_t (the model output at time step t): $\tilde{\ell}(\theta; d_t) = \langle \nabla_\theta \ell(\theta_t; d_t), \theta - \theta_t \rangle$, ii) Under appropriate choice of λ, optimizing for $\theta_{t+1} = \arg\min_{\theta \in \mathcal{C}} \sum_{i=1}^{t} \tilde{\ell}(\theta; d_t) + \frac{\lambda}{2} \|\theta\|_2^2$ over $\theta \in \mathcal{C}$ gives a good model at step $t+1$, and iii) For all $t \in [n]$, one can privately keep track of $\sum_{i=1}^{t} \tilde{\ell}(\theta; d_t)$ using the *tree aggregation protocol* [DNPR10; CSS11] described above.

In Theorem 6.14, we provide the privacy guarantee for Algorithm 4. The proof is immediate from Theorem 6.13.

Theorem 6.14 (Privacy guarantee). *If the noise multiplier $\lambda = \frac{\sqrt{2\lceil \lg(n) \rceil (\ln(1/\delta)+\varepsilon)}}{\varepsilon}$, then Algorithm 4 (Algorithm $\mathcal{A}_{\text{FTRL}}$) guarantees (ε, δ)-differential privacy.*

The theorem here gives a regret guarantee for Algorithm 4 against a *fully adaptive* [Sha+11] adversary who chooses the loss function $\ell(\theta; d_t)$ based on $[\theta_1, \ldots, \theta_t]$, but without knowing the internal randomness of the algorithm.

Algorithm 4 $\mathcal{A}_{\text{FTRL}}$: Differentially Private Follow-The-Regularized-Leader (DP-FTRL)

Require: Data set: $D = \{d_1, \cdots, d_n\}$ arriving in a stream, in an arbitrary order; constraint set: $\mathcal{C}$, noise multiplier: λ, regularization parameter: Δ, clipping norm: L.

1: $\theta_1 \leftarrow \arg\min_{\theta \in \mathcal{C}} \frac{\Delta}{2} \|\theta\|_2^2$. **Output** θ_1.
2: $\mathcal{T} \leftarrow \mathsf{InitializeTree}\,(n, \sigma^2, L)$.
3: **for** $t \in [n]$ **do**
4: Let $\nabla_t \leftarrow \text{clip}\,(\nabla_\theta \ell(\theta_t; d_t), L)$, where $\text{clip}\,(v, L) = v \cdot \min\left\{\frac{L}{\|v\|_2}, 1\right\}$, taking $\nabla_\theta \ell(\theta; \perp) = \mathbf{0}$.
5: $\mathcal{T} \leftarrow \mathsf{AddToTree}\,(\mathcal{T}, t, \nabla_t)$.
6: $s_t \leftarrow \mathsf{GetSum}\,(\mathcal{T}, t)$, i.e., estimate $\sum_{i=1}^t \nabla_i$ via tree-aggregation protocol.
7: $\theta_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \langle s_t, \theta \rangle + \frac{\Delta}{2} \|\theta\|_2^2$. **Output** θ_{t+1}.
8: **end for**

Theorem 6.15 (Regret guarantee)**.** *Let $\{\theta_1, \ldots, \theta_n\}$ be the outputs of Algorithm $\mathcal{A}_{\text{FTRL}}$ (Algorithm 4), and L be a bound on the ℓ_2-Lipschitz constant of the loss functions. W.p. at least $1 - \beta$ over the randomness of $\mathcal{A}_{\text{FTRL}}$, the following is true for any $\theta^* \in \mathcal{C}$.*

$$\frac{1}{n}\sum_{t=1}^n \ell(\theta_t; d_t) - \frac{1}{n}\sum_{t=1}^n \ell(\theta^*; d_t) \leq \frac{L\lambda\sqrt{p\lceil \lg n \rceil \ln(n/\beta)} + L^2}{\Delta} + \frac{\Delta}{2}\left(\|\theta^*\|_2^2 - \|\theta_1\|_2^2\right).$$

Setting Δ optimally and plugging in the noise multiplier λ from Theorem 6.14 to ensure (ε, δ)-differential privacy, we have

$$R_D(\mathcal{A}_{\text{FTRL}}; \theta^*) = O\left(L\|\theta^*\|_2 \sqrt{n} \cdot \left(1 + \sqrt{\frac{p^{1/2}(\ln^2(1/\delta) + \varepsilon)\ln(1/\beta)}{\varepsilon}}\right)\right).$$

Proof. Recall that by Algorithm $\mathcal{A}_{\text{FTRL}}$, $\theta_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \underbrace{\sum_{i=1}^t \langle \nabla_i, \theta \rangle + \frac{\Delta}{2}\|\theta\|_2^2 + \langle b_t, \theta \rangle}_{J_t^{\text{priv}}(\theta)}$, where the Gaussian noise $b_t = s_t - \sum_{i=1}^t \nabla_i$ for s_t being the output of $\mathsf{GetSum}\,(\mathcal{T}, t)$. By standard concentration of spherical Gaussians, w.p. at least $1 - \beta$, $\forall t \in [n]$, $\|b_t\|_2 \leq L\lambda\sqrt{p\lceil \lg(n) \rceil \ln(n/\beta)}$. We will use this

bound to control the error introduced due to privacy. Now, consider the optimizer of the non-private objective:

$$\widetilde{\theta}_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \underbrace{\sum_{i=1}^{t} \langle \nabla_i, \theta \rangle + \frac{\Delta}{2} \|\theta\|_2^2}_{f_t^{\mathrm{np}}(\theta)}, \qquad \text{where } \nabla_t = \nabla \ell(\theta_t; d_t).$$

That is, post-hoc we consider the hypothetical application of non-private FTRL to the same sequence of *linearized* loss functions $f_t(\widetilde{\theta}) = \langle \nabla_t, \widetilde{\theta} \rangle = \langle \nabla \ell(\theta_t; d_t), \widetilde{\theta} \rangle$ seen in the private training run. In the following, we will first bound how much the models output by $\mathcal{A}_{\mathrm{FTRL}}$ deviate from models output by the hypothetical non-private FTRL discussed above. Then, we invoke standard regret bound for FTRL, while accounting for the deviation of the models output by $\mathcal{A}_{\mathrm{FTRL}}$. By the same calculation as in (6.29) in Section 6.2.2, we obtain

$$\left\| \widetilde{\theta}_{t+1} - \theta_{t+1} \right\|_2 \leq \frac{\|b_t\|_2}{\Delta}. \tag{6.56}$$

We can now easily bound the regret. By standard linear approximation "trick" from the online learning literature [Sha12; Haz19], we have the following. For $\nabla_t = \nabla_\theta \ell(\theta_t; d_t)$,

$$\begin{aligned} \sum_{t=1}^{n} \ell(\theta_t; d_t) - \sum_{t=1}^{n} \ell(\theta^*; d_t) &\leq \sum_{t=1}^{n} \langle \nabla_t, \theta_t - \theta^* \rangle \\ &= \sum_{t=1}^{n} \langle \nabla_t, \theta_t - \widetilde{\theta}_t + \widetilde{\theta}_t - \theta^* \rangle \\ &= \underbrace{\sum_{t=1}^{n} \langle \nabla_t, \widetilde{\theta}_t - \theta^* \rangle}_{A} + \underbrace{\sum_{t=1}^{n} \langle \nabla_t, \theta_t - \widetilde{\theta}_t \rangle}_{B}. \end{aligned} \tag{6.57}$$

One can bound the term A in (6.57) by Theorem 5.2 of [Haz19] and get $A \leq \left(\frac{L^2 n}{\Delta} + \frac{\Delta}{2} \left(\|\theta^*\|_2^2 - \|\theta_1\|_2^2 \right) \right)$. As for term B, using (6.56) and the concentration on b_t mentioned earlier, we have, w.p. at least $1 - \beta$,

$$B \leq \sum_{t=1}^{n} \|\nabla_t\|_2 \cdot \left\| \widetilde{\theta}_t - \theta_t \right\|_2 \leq \sum_{t=1}^{n} L \cdot \left\| \widetilde{\theta}_t - \theta_t \right\|_2 \leq \frac{2L\lambda \sqrt{p \lceil \lg n \rceil \ln(n/\beta)}}{\Delta}. \tag{6.58}$$

Combining (6.57) and (6.58), we immediately have the first part of the theorem. To prove the second part, we just optimize for the regularization parameter Δ and plug in the noise multiplier λ from Theorem 6.14. □

Translating Theorem 6.15 Into Excess Empirical Risk

As we saw in (6.54), one can interpret the regret guarantee as a bound on the excess empirical risk. This essentially means the excess empirical risk one can obtain by using Algorithm $\mathcal{A}_{\text{FTRL}}$ is $O\left(\frac{p^{1/4}\sqrt{n}\ln^2(1/\delta)}{\sqrt{\varepsilon}} \cdot \text{polylog}\,(n)\right)$. While this bound is strictly worse than the one we obtained for Algorithm $\mathcal{A}_{\text{DP-SGD}}$ via Theorem 6.11, the oracle complexity of $\mathcal{A}_{\text{FTRL}}$ is better by a factor of n. A natural question that arises is whether, we can have $\mathcal{A}_{\text{FTRL}}$ achieve similar utility guarantee as $\mathcal{A}_{\text{DP-SGD}}$ under similar oracle complexity. This is still an open question for research.

Practical Extensions

One of the major advantages of DP-FTRL over DP-SGD is that it can operate over a stream of data samples, while still providing strong privacy/utility trade-offs. This makes it an attractive choice for settings like federated learning (FL), where there isn't a single static data set in a centralized location. However, extending DP-FTRL to settings require a much more involved privacy analysis as one user can contribute multiple training examples, during the training process. See [Kai+21a] for a detailed discussion.

6.4 DP Empirical Risk Minimization with ℓ_1/ℓ_∞-geometry

The algorithms presented till now best work when the objective function is Lipschitz with respect to ℓ_2-norm. But in many machine learning tasks, especially those with sparsity constraint, the objective function is often Lipschitz with respect to ℓ_1-norm. For example, in the high-dimensional linear regression setting e.g. the classical LASSO algorithm [Tib96], we would like to compute $\arg\min_{\|\theta\|_1 \le s} \|X\theta - y\|_2^2$. In the usual case of $|x_{ij}| = O(1), |y_i| = O(s)$, the loss function $\ell(\theta; (x_i, y_i)) = \|y_i - \langle x_i, \theta\rangle\|_2^2$ is $O(s)$-Lipschitz with respect to ℓ_1-norm but is $O(s\sqrt{p})$-Lipschitz with respect to ℓ_2-norm within the constraint set $\|\theta\|_1 \le s$. Here x_i correspond to the i-th row of the design matrix X, and y_i corresponds to the i-th coordinate of the response vector y.

Let us consider the data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the loss function $\mathcal{L}(\theta; D) = \sum_{i=1}^n (y_i - \langle x_i, \theta\rangle)^2$, and the constraint set $\mathcal{C}$ to be $\|\theta\|_1 \le s$. Then applying any of the algorithms mentioned earlier on the corresponding ERM problem would result in an excess empirical risk of $\widetilde{O}_{\varepsilon,\delta}\left(s^2\sqrt{p}\right)$. For the interesting high-dimensional parameter regime, where $s \ll n \ll p$, this bound is vacuous. We

would ideally want an excess empirical risk of the form $O_{\varepsilon,\delta}\left(\text{poly}\left(s, \ln(p)\right)\right)$. Here, $\widetilde{O}_{\varepsilon,\delta}(\cdot)$ hides the privacy parameters ε and δ.

In this section, we will show that in the high-dimensional setting it is more effective to use the private version of the classical Frank-Wolfe algorithm [FW56]. In particular, we show that for LASSO, such algorithm achieves the nearly optimal privacy risk of $\Theta_{\varepsilon,\delta}(s^2 \cdot \text{polylog}\left(p\right) \cdot n^{1/3})$.

6.4.1 Frank-Wolfe Algorithm

We present the algorithm in this section as a purely optimization procedure that minimizes a convex function $f : \mathcal{C} \to \mathbb{R}$. The Frank-Wolfe algorithm [FW56] can be regarded as a "greedy" algorithm which moves towards the optimum solution in the first order approximation (see Algorithm 5 for the description). How fast Frank-Wolfe algorithm converges depends f's "curvature", defined as follows according to [Cla10; Jag13]. We remark that a γ -smooth function on $\mathcal{C}$ has curvature constant bounded by $\gamma \ \|\mathcal{C}\|_2^2$.

Definition 6.16 (Curvature constant)**.** *For $f : \mathcal{C} \to \mathbb{R}$, define Γ_f as below.*

$$\Gamma_f := \sup_{\theta_1,\theta_2,\in\mathcal{C},\gamma\in(0,1],\theta_3=\theta_1+\gamma(\theta_2-\theta_1)} \frac{2}{\gamma^2}\left(f(\theta_3) - f(\theta_1) - \langle\theta_3 - \theta_1, \nabla f(\theta_1)\rangle\right).$$

Remark 6.17. *One can show ([Cla10; Jag13]) that for any $q, r \geq 1$ such that $q^{-1} + r^{-1} = 1$, Γ_f is upper bounded by $\lambda\|\mathcal{C}\|_q^2$, where $\lambda = \max_{\theta\in\mathcal{C},\|v\|_q=1} \|\nabla^2 f(\theta) \cdot v\|_r$.*

Remark 6.18. *One useful bound is for the quadratic programming $f(\theta) = \theta^\top X^T X\theta + \langle b, \theta\rangle$. In this case, by [Cla10], $\Gamma_f \leq \max_{a,b\in X\cdot\mathcal{C}} \|a - b\|_2^2$. When $\mathcal{C}$ is centrally symmetric, we have the bound $\Gamma_f \leq 4\max_{\theta\in\mathcal{C}} \|X\theta\|_2^2$.*

Algorithm 5 Frank-Wolfe algorithm

Require: $\mathcal{C} \subseteq \mathbb{R}^p, f : \mathcal{C} \to \mathbb{R}, \mu$

1: Choose an arbitrary θ_1 from $\mathcal{C}$;
2: **for** $t = 1$ to $T - 1$ **do**
3: Compute $\widehat{\theta}_t = \arg\min_{\theta\in\mathcal{C}}\langle\nabla f(\theta_t), \theta - \theta_t\rangle$;
4: Set $\theta_{t+1} = \theta_t + \mu(\widehat{\theta}_t - \theta_t)$;
5: **end for**
6: return θ_T.

Define $\theta^* = \arg\min_{\theta\in\mathcal{C}} f(\theta)$. The following theorem shows the convergence of Frank-Wolfe algorithm.

Theorem 6.19 ([Cla10; Jag13]). *If we set* $\mu = 1/T$, *then* $f(\theta_T) - f(\theta^*) = O(\Gamma_f/T)$.

While the Frank-Wolfe algorithm does not necessarily provide faster convergence compared to the gradient-descent based method, it has two major advantages. First, on Line 3, it reduces the problem to solving a minimization of linear function. When $\mathcal{C}$ is defined by small number of vertices, e.g. when $\mathcal{C}$ is an ℓ_1-ball, the minimization can be done by checking $\langle \nabla f(\theta_t), x\rangle$ for each vertex x of $\mathcal{C}$. This can be done efficiently. Secondly, each step in Frank-Wolfe takes a convex combination of θ_t and $\widehat{\theta}_t$, which is on the boundary of $\mathcal{C}$. Hence each intermediate solution is always inside $\mathcal{C}$ (sometimes called *projection free*), and the final outcome θ_T is the convex combination of up to T points on the boundary of $\mathcal{C}$ (or vertices of $\mathcal{C}$ when $\mathcal{C}$ is a polytope). Such outcome might be desired, for example when $\mathcal{C}$ is a polytope, as it corresponds to a sparse solution. Due to these reasons Frank-Wolfe algorithm has found many applications in machine learning [SSZ10; HK12; Cla10]. As we shall see below, these properties are also useful for obtaining low risk bounds for their private version.

6.4.2 Private Frank-Wolfe Algorithm

There are different ways to make Algorithm 5 private, dependent on the geometry of $\mathcal{C}$. Here we focus on the important case where $\mathcal{C}$ is a polytope, corresponding to the LASSO problem. In this case, we apply the exponential mechanism [MT07] to achieve privacy. We now present a private version of the Frank-Wolfe algorithm. We can achieve privacy by replacing Line 3 in Algorithm 5 with its private version in one of two ways. In the first variant, we can apply exponential mechanism [MT07] to guarantee privacy; and in the second variant, we can apply objective perturbation (Algorithm $\mathcal{A}_{\text{obj}-\text{pert}}$) or DP-SGD (Algorithm $\mathcal{A}_{\text{DP}-\text{SGD}}$) or DP-FTRL (Algorithm $\mathcal{A}_{\text{FTRL}}$). The first variant works especially well when $\mathcal{C}$ is a polytope defined by polynomially many vertices. In this case, we show that the error depends on the ℓ_1-Lipschitz constant, which can be much smaller than the ℓ_2-Lipschitz constant. In particular, the private Frank-Wolfe algorithm is nearly optimal for the important high-dimensional sparse linear regression (or compressive sensing) problem. For the details on the second variant, see [TTZ15].

Algorithm 6 describes the private version of Frank-Wolfe algorithm for the polytope case, i.e. when $\mathcal{C}$ is a convex hull of a finite set S of vertices (or corners). In this case, we know that any linear function is minimized at one point of S per the following basic fact.

Fact 6.20. *Let* $\mathcal{C} \subseteq \mathbb{R}^p$ *be the convex hull of a compact set* $S \subseteq \mathbb{R}^p$. *For any vector* $v \in \mathbb{R}^p$, $\arg\min_{\theta\in\mathcal{C}} \langle \theta, v\rangle \cap S \neq \emptyset$.

Since θ_{t+1} can be selected as one of $|S|$ vertices, by applying the exponential mechanism [MT07], we obtain differentially private algorithm with risk logarithmically dependent on $|S|$. When $|S|$ is polynomial in p, it leads to an error bound with $\ln p$ dependence. While the exponential mechanism can be applied to the general $\mathcal{C}$ as well, its error would depend on the size of a cover of the boundary of $\mathcal{C}$, which can be exponential in p, leading to an error bound with polynomial dependence on p.

Algorithm 6 $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$: Differentially Private Frank-Wolfe Algorithm (Polytope Case)

Require: Data set: $D = \{d_1, \cdots, d_n\}$, loss function: $\mathcal{L}(\theta; D) = \sum_{i=1}^{n} \mathcal{L}(\theta; d_i)$ (with ℓ_1-Lipschitz constant L_1 for ℓ), privacy parameters: (ε, δ), convex set: $\mathcal{C} = conv(S)$ with $\|\mathcal{C}\|_1$ denoting $\max_{s \in S} \|s\|_1$ and S being the set of corners.

1: Choose an arbitrary θ_1 from $\mathcal{C}$;
2: **for** $t = 1$ to $T - 1$ **do**
3: $\quad \forall s \in S, \alpha_s \leftarrow \langle s, \nabla \mathcal{L}(\theta_t; D) \rangle + \text{Lap}\left(\frac{L_1 \|\mathcal{C}\|_1 \sqrt{8T \ln(1/\delta)}}{n\varepsilon}\right)$, where $\text{Lap}(\lambda) \sim \frac{1}{2\lambda} e^{-|x|/\lambda}$.
4: $\quad \widehat{\theta_t} \leftarrow \arg\min_{s \in S} \alpha_s$.
5: $\quad \theta_{t+1} \leftarrow (1 - \mu)\theta_t + \mu \widehat{\theta_t}$, where $\mu = \frac{1}{T+2}$.
6: **end for**
7: Output $\theta^{\text{priv}} = \theta_T$.

Theorem 6.21 (Privacy guarantee). *Algorithm 6 (Algorithm $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$) is (ε, δ)-differentially private.*

The proof of privacy follows from a straight forward use of exponential mechanism [MT07; BLST10] (the noisy maximum version from [BLST10], Theorem 5) and the advanced composition theorem discussed earlier in the book. In Theorem 6.22 we prove the utility guarantee for the private Frank-Wolfe algorithm for the convex polytope case. Define $\Gamma_\ell = \max_{d \in \mathcal{D}} C_\ell(d)$ over all the possible data sets in the domain.

Theorem 6.22 (Utility guarantee). *Let L_1,S and $\|\mathcal{C}\|_1$ be defined as in Algorithms 6 (Algorithm $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$). Let Γ_ℓ be an upper bound on the curvature constant (defined in Definition 6.16) for the loss function $\ell(\cdot; d)$ that holds for all $d \in \mathcal{D}$. In Algorithm $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$, if we set $T = \frac{{\Gamma_\ell}^{2/3} (n\varepsilon)^{2/3}}{(L_1 \|\mathcal{C}\|_1)^{2/3}}$, then*

$$\mathbf{E}\left[\mathcal{L}(\theta^{priv}; D)\right] - \min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; D)$$

$$= O\left(\frac{{\Gamma_\ell}^{1/3}\,(L_1\,\|\mathcal{C}\|_1)^{2/3}\,n^{1/3}\ln(n|S|)\sqrt{\ln(1/\delta)}}{\varepsilon^{2/3}}\right).$$

Here the expectation is over the randomness of the algorithm.

Proof. For ease of notation we hide the dependence of $\mathcal{L}$ on the data set D and represent it simply as $\mathcal{L}(\theta)$. In order to prove the utility guarantee we first invoke the utility guarantee of the non-private noisy Frank-Wolfe algorithm from [Jag13] (Theorem 1).

Theorem 6.23 (Non-private utility guarantee [Jag13])**.** *Assume the conditions in Theorem 6.22 and let $\gamma > 0$ be fixed. Recall that $\mu = 1/(T+2)$ and let $\phi_1 \in \mathcal{C}$. Suppose that $\langle s_1, \cdots, s_T\rangle$ is a sequence of vectors from $\mathcal{C}$, with $\phi_{t+1} = (1-\mu)\phi_t + \mu s_t$ such that for all $t \in [T]$,*

$$\langle s_t, \nabla\mathcal{L}(\phi_t)\rangle \le \min_{s\in\mathcal{C}}\langle s, \nabla\mathcal{L}(\phi_t)\rangle + \frac{1}{2}\gamma\,\mu(n\cdot\Gamma_\ell).$$

Then,

$$\mathcal{L}(\phi_T) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta) \le \frac{2n\cdot\Gamma_\ell}{T+2}\,(1+\gamma)\,.$$

Since the convex set $\mathcal{C}$ is a polytope with corners in S, if s_t in Theorem 6.23 corresponds to $\widehat{\theta_t}$ in Algorithm $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$, and ϕ_t corresponds to θ_t in $\mathcal{A}_{\text{Noise}-\text{FW(polytope)}}$, then using the tail properties of Laplace distribution and Fact 6.20 one can show that with probability at least $1-\beta$, the term γ in Theorem 6.23 is at most $O\left(\frac{L_1\|\mathcal{C}\|_1\sqrt{8T\ln(1/\delta)}\ln(|S|T/\beta)}{\mu\varepsilon(n\cdot\Gamma_\ell)}\right)$. Plugging in this bound in Theorem 6.23, we immediately get that with probability at least $1-\beta$,

$$\mathcal{L}(\theta_T) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta) = O\left(\frac{\Gamma_{\mathcal{L}}}{T} + \frac{L_1\,\|\mathcal{C}\|_1\,\sqrt{8T\ln(1/\delta)}\ln(|S|T/\zeta)}{\varepsilon}\right). \quad (6.59)$$

From, (6.59) we can conclude the following in expectation.

$$\mathbf{E}\left[\mathcal{L}(\theta_T) - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta)\right]$$
$$= O\left(\frac{n\cdot\Gamma_\ell}{T} + \frac{L_1\,\|\mathcal{C}\|_1\,\sqrt{8T\ln(1/\delta)}\ln(TL_1\,\|\mathcal{C}\|_1\cdot|S|)}{\varepsilon}\right). \quad (6.60)$$

Setting $T = \frac{{\Gamma_\ell}^{2/3}(n\varepsilon)^{2/3}}{(L_1\|\mathcal{C}\|_1)^{2/3}}$ results in the claimed utility guarantee. □

6.4.3 Nearly Optimal Private LASSO

We now apply the private Frank-Wolfe algorithm $\mathcal{A}_{\text{Noise}-\text{FW}(\text{polytope})}$ to the important case of the sparse linear regression (or LASSO) problem. We show that the private Frank-Wolfe algorithm leads to a nearly tight $O_{\varepsilon,\delta}(n^{1/3} \cdot \text{polylog}(p))$ bound. $O_{\varepsilon,\delta}(\cdot)$ hides terms in the privacy parameters.

Problem Definition

Given a data set $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ of n-samples from the domain $D = \{(x, y) : x \in \mathbb{R}^p, y \in [-1, 1], \|x\|_\infty \leq 1\}$, and the convex set $\mathcal{C} = \ell_1^p$. Define the squared loss,

$$\mathcal{L}(\theta; D) = \frac{1}{2} \sum_{i \in [n]} (\langle x_i, \theta \rangle - y_i)^2 . \tag{6.61}$$

The objective is to compute $\theta^{\text{priv}} \in \mathcal{C}$ to minimize $\mathcal{L}(\theta; D)$ while preserving privacy with respect to add/removal of individual (x_i, y_i) pair. The non-private setting of the above problem is a variant of the least squares problem with ℓ_1 regularization, which was started by the work of LASSO [Tib96; Tib+97] and intensively studied in the past years [HTF01; DJ04; CT05; Don06; CT07; BRT09; BM12; RWY09; Zha13]. One important reason for using ℓ_1 regularization is to induce sparse solutions, i.e. θ with small number of non-zero coordinates. This is especially interesting for the so called "high-dimensional" setting where $p \gg n$. Indeed, via a long line of work [DJ04; CT05; Don06; Wai06; CT07; BRT09], it has been shown that under suitable condition of X, using ℓ_1 regularization can indeed produce a nearly optimal sparse solution, providing theoretical support to the empirical success of LASSO.

Since the ℓ_1 ball is the convex hull of $2p$ vertices, we can apply the private Frank-Wolfe algorithm $\mathcal{A}_{\text{Noise}-\text{FW}(\text{polytope})}$. For the above setting, it is easy to check that the ℓ_1-Lipschitz constant is bounded by $O(1)$. Further, by applying the bound on quadratic programming Remark 6.18, we have that $C_\ell \leq 4 \max_{\theta \in \mathcal{C}, \|x\|_\infty \leq 1} \langle x, \theta \rangle^2 = O(1)$ since $\mathcal{C}$ is the unit ℓ_1 ball, and $|x_{ij}| \leq 1$. Hence $\Gamma_\ell = O(1)$. Now applying Theorem 6.22, we have

Corollary 6.24. *Let $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ of n samples from the domain $D = \{(x, y) : \|x\|_\infty \leq 1, |y| \leq 1\}$, and the convex set $\mathcal{C}$ equal to the ℓ_1-ball. The output θ^{priv} of Algorithm $\mathcal{A}_{\text{Noise}-\text{FW}(\text{polytope})}$ ensures the following.*

$$\mathbf{E}\left[\mathcal{L}(\theta^{\text{priv}}; D)\right] - \min_{\theta \in \mathcal{C}} \mathcal{L}(\theta; D) = O\left(\frac{n^{1/3} \ln(np/\delta)}{\varepsilon^{2/3}}\right).$$

Comparison to Algorithms for Private Sparse Support Selection

For a high-dimensional linear regression problem of the form $y = \langle x, \theta^* \rangle +$ noise with $\|\theta^*\|_0 = O(1)$, Algorithms in [KST12; ST13b] allow one to identify the non-zero coordinates of $]\theta^*$ exactly under (ε, δ)-DP. It is not hard to observe that this is a much tighter guarantee as opposed to Corollary 6.24. However, the algorithms in [KST12; ST13b] operate under much stronger assumptions like *restricted strong convexity* or *mutual incoherence* [Wai06]. These assumptions are known to be necessary even for non-private support selection, and may be too hard to satisfy in practice [Was12].

Note on the Lower Bound

As mentioned earlier, the bound obtained via Corollary 6.24 is essentially tight. The proof of the lower bound follows the standard template of fingerprinting codes, use to prove lower bounds for (ε, δ)-DP [Vad17; TTZ15].

Oracle Complexity

Algorithm $\mathcal{A}_{\text{Noise}-\text{FW}(\text{polytope})}$ essentially makes $O(T \cdot n)$ oracle calls to obtain the tight privacy/utility trade-off. Compared to non-private Frank-Wolfe, the oracle complexity remain unchanged up to terms depending on the privacy parameters (ε, δ), and additional poly-logarithmic factors. This is a common theme we saw in all the algorithms in this chapter. Differential privacy tends not hurt the rate of convergence. However, it only allows convergence to a specific error level allowed by the privacy constraints.

6.5 Lower Bounds, and Algorithms not Considered

6.5.1 Lower Bounds on Private Constrained Optimization

In this section, we discuss at a high-level some of the standard lower bounding techniques used for proving optimality of DP optimization algorithms. The lower bounds for the algorithms considered in this chapter are in some form dependent on the lower bound for estimating one-way marginals with DP (Theorem 6.25 below.)

Theorem 6.25 (Lower bounds for 1-way marginals)**.**

1. ***ε-differential private algorithms:*** *Let $n, p \in \mathbb{N}$ and $\varepsilon > 0$. There is a number $M = \Omega\left(\min\left(n, p/\varepsilon\right)\right)$ such that for every ε-differentially private algorithm $\mathcal{A}$, there is a dataset $D = \{d_1, \ldots, d_n\} \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ with $\left\|\sum_{i=1}^n d_i\right\|_2 \in [M-1, M+1]$ such that, with probability at least $1/2$ (taken over the algorithm*

random coins), we have

$$\left\|\mathcal{A}(D) - q(D)\right\|_2 = \Omega\left(\min\left(1, \frac{p}{\varepsilon n}\right)\right),$$

where $q(D) = \frac{1}{n}\sum_{i=1}^n d_i$.

2. (ε, δ)***-differential private algorithms:*** *Let* $n, p \in \mathbb{N}$, $\varepsilon > 0$, *and* $\delta = o(\frac{1}{n})$. *There is a number* $M = \Omega\left(\min\left(n, \sqrt{p}/\varepsilon\right)\right)$ *such that for every* (ε, δ)*-differentially private algorithm* $\mathcal{A}$, *there is a dataset* $D = \{d_1, \ldots, d_n\} \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ *with* $\left\|\sum_{i=1}^n d_i\right\|_2 \in [M-1, M+1]$ *such that, with probability at least* $1/3$ *(taken over the algorithm random coins), we have*

$$l\left\|\mathcal{A}(D) - q(D)\right\|_2 = \Omega\left(\min\left(1, \frac{\sqrt{p}}{\varepsilon n}\right)\right),$$

where $q(D) = \frac{1}{n}\sum_{i=1}^n d_i$.

Now define a dataset $D = \{d_1, \ldots, d_n\}$ with data points drawn from $\left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$, and any $\theta \in \mathbb{B}$, define $\mathcal{L}(\theta; D) = -\langle \theta, \sum_{i=1}^n d_i\rangle$. Clearly, $\mathcal{L}$ is linear and, hence, Lipschitz and convex. Note that, whenever $\left\|\sum_{i=1}^n d_i\right\|_2 > 0$, $\theta^* = \frac{\sum_{i=1}^n d_i}{\left\|\sum_{i=1}^n d_i\right\|_2}$ is the minimizer of $\mathcal{L}(\theta; D)$ over $\mathbb{B}$. Next, we show lower bounds on the excess empirical risk incurred by any ε and (ε, δ) differentially private algorithm with output $\theta^{\mathsf{priv}} \in \mathbb{B}$.

Theorem 6.26 (Lower bound for ε-differentially private algorithms)**.** *Let* $n, p \in \mathbb{N}$ *and* $\varepsilon > 0$. *For every* ε*-differentially private algorithm (whose output is denoted by* θ^{priv}*), there is a dataset* $D = \{d_1, \ldots, d_n\} \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ *such that, with probability at least* $1/2$ *(over the algorithm random coins), we must have*

$$\mathcal{L}(\theta; D) - \mathcal{L}(\theta^*; D) = \Omega\left(\min\left(n, p/\varepsilon\right)\right),$$

where $\theta^* = \frac{\sum_{i=1}^n d_i}{\left\|\sum_{i=1}^n d_i\right\|_2}$ *is the minimizer of* $\mathcal{L}(\theta; D)$ *over* $\mathbb{B}$ *and* $\mathcal{L}$ *is defined above.*

Proof. Let $\mathcal{A}$ be an ε-differentially private algorithm for minimizing $\mathcal{L}$ and let θ^{priv} denote its output. First, observe that for any $\theta \in \mathbb{B}$ and dataset D, $\mathcal{L}(\theta; D) - \mathcal{L}(\theta^*; D) = \left\|\sum_{i=1}^n d_i\right\|_2 (1 - \langle \theta, \theta^*\rangle)$. Hence, we have $\mathcal{L}(\theta; D) - \mathcal{L}(\theta^*; D) \geq \frac{1}{2}\left\|\sum_{i=1}^n d_i\right\|_2 \|\theta - \theta^*\|_2^2$. This is due to the fact that $\|\theta - \theta^*\|_2^2 = \|\theta^*\|_2^2 + \|\theta\|_2^2 - 2\langle \theta, \theta^*\rangle$ and the fact that $\theta^*, \theta \in \mathbb{B}$.

Let $M = \Omega\left(\min\left(n, p/\varepsilon\right)\right)$ be as in Part 1 of Theorem 6.25. Suppose, for the sake of a contradiction, that for every dataset $D \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ with $\left\|\sum_{i=1}^n d_i\right\|_2 \in [M-1, M+1]$, with probability more than $1/2$, we have

$\left\|\theta^{\text{priv}} - \theta^*\right\|_2 \neq \Omega(1)$. Let $\tilde{\mathcal{A}}$ be an ε-DP algorithm that first runs $\mathcal{A}$ on the data and then outputs $\frac{M}{n}\theta^{\text{priv}}$. Note that this implies that for every data set $D \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ with $\left\|\sum_{i=1}^n d_i\right\|_2 \in [M-1, M+1]$, with probability more than $1/2$, $\left\|\tilde{\mathcal{A}}(D) - q(D)\right\|_2 \neq \Omega\left(\min\left(1, \frac{p}{\varepsilon n}\right)\right)$ which contradicts Part 1 of Theorem 6.25. Thus, there must exist a dataset $D \subseteq \left\{-\frac{1}{\sqrt{p}}, \frac{1}{\sqrt{p}}\right\}^p$ with $\left\|\sum_{i=1}^n d_i\right\|_2 = \Omega\left(\min\left(n, p/\varepsilon\right)\right)$ such that with probability at least $1/2$, we have $\left\|\theta^{\text{priv}} - \theta^*\right\|_2 = \Omega(1)$. Therefore, from the observation we made in the previous paragraph, we have, with probability at least $1/2$, $\mathcal{L}(\theta^{\text{priv}}; D) - \mathcal{L}(\theta^*; D) = \Omega\left(\min\left(n, p/\varepsilon\right)\right)$. □

The lower bound for the (ε, δ)-DP case follows by the same argument as in Theorem 6.26, but reducing the problem instance to Part 2 in Theorem 6.25. The lower bound for the setting in Section 6.4 follows from a slightly complicated variant of Theorem 6.25. We encourage the readers to read [TTZ15] for more details.

6.5.2 Algorithms not Considered

In the exposition of this chapter, we left out quite a few important algorithms, primarily for the ease of presentation. In this section, we highlight some of them, and encourage interested readers to explore more.

Additional Assumptions on the Loss Function

In this chapter, we primarily focused on obtaining excess empirical risk bounds for Lipschitz convex convex functions. For the ℓ_2-Lipschitz setting, if we additionally allow the loss functions to be Δ-strongly convex, then the excess empirical risk for the pure ε-case improves to $O\left(\frac{L^2 p^2 \log(n)}{n\Delta\varepsilon^2}\right)$. The algorithm is a two stage-algorithm, with a first localization step to identify a small set where the true minimizer lies, and then running Algorithm $\mathcal{A}_{\text{exp}-\text{samp}}$ on that set [BST14]. In the (ε, δ)-case, the corresponding excess empirical risk becomes $O\left(\frac{L^2 p \cdot \text{polylog}(n/\delta)}{n\Delta\varepsilon^2}\right)$. The algorithms that achieve this bound are mild variants of Algorithms $\mathcal{A}_{\text{DP}-\text{SGD}}$ and $\mathcal{A}_{\text{obj}-\text{pert}}$. Assuming smoothness however, does not improve the excess empirical risk. However, it improves the rate of convergence/oracle complexity to achieve the same error.

Algorithms for Optimal Population Risk

As mentioned in the beginning of the chapter that we would focus only on excess empirical risk, and for estimating the excess true/population risk we would

use (6.3). While this is a natural way to translate from excess empirical risk to excess population risk, this translation does not result in optimal excess population risk. For ℓ_2-Lipschitz convex functions (with Lipschitz constant $O(1)$, and for a constraint set with $\|\mathcal{C}\|_2 = O(1)$) the optimal population risk is $O\left(\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{p \cdot \ln(1/\delta)}}{\varepsilon n}\right) \cdot \text{polylog}(n)\right)$, and for 1-strongly convex functions it is $O\left(\left(\frac{1}{n} + \frac{p \cdot \ln(1/\delta)}{\varepsilon^2 n^2}\right) \cdot \text{polylog}(n)\right)$. Notice that in both these cases, the lower-order term in n only depends on the privacy parameter, as opposed to the one obtained via (6.3). The algorithms that achieve these bounds are variants of Algorithm $\mathcal{A}_{\text{DP-SGD}}$ or Algorithm $\mathcal{A}_{\text{obj-pert}}$, with substantially more involved utility analysis, using tools from algorithmic stability. For a detailed discussion on these techniques, see [BFGT20]. When the geometry is ℓ_1/ℓ_∞ (as in Section 6.4), the corresponding optimal excess population risk of $O\left(\sqrt{\frac{\ln(p)}{n}} + \frac{\sqrt{\ln(p)\ln(1/\delta)}}{(n\varepsilon)^{2/3}}\right)$ is obtained a variant of Algorithm $\mathcal{A}_{\text{Noise-FW(polytope)}}$. This bound assumes all the assumptions in Corollary 6.24. For a detailed exposition to this result see [AFKT21].

Other Algorithms not Discussed

For the purposes of brevity, we left out the discussion of a number of algorithms. In particular, we did not mention any algorithm which are optimal under local differential privacy [STU17]. We also did not discuss algorithms like *private cutting plane methods* [STU17], *output perturbation* [CMS11], *bolt-on privacy* [Wu+17], *private mirror descent* [TTZ14b], *privacy amplification by iteration* [FMTT18]. Many of these algorithms have advantages over the algorithms mentioned in this chapter, under specific problem settings. We encourage the readers to explore these algorithms, and their impact on the broader space of private constrained optimization.

In this chapter, we focused only on convex loss functions. As mentioned earlier algorithms like $\mathcal{A}_{\text{FTRL}}$, $\mathcal{A}_{\text{DP-SGD}}$, and $\mathcal{A}_{\text{Noise-FW(polytope)}}$ are applicable to non-convex losses too, as their privacy do not depend on convexity. While there are restricted utility analyses for non-convex losses [STT20; WCX19], it is still an active area of research. Additionally, there are better algorithms known specific to problems like linear regression [STU17; She19], or principal component analysis [DTTZ14; LMV21]. We did not explore these problems in this chapter.

Acknowledgements

Most of the presentation in this chapter is based off prior work, unless mentioned explicitly. The privacy/utility analysis of Algorithms $\mathcal{A}_{\text{exp-samp}}$ and $\mathcal{A}_{\text{DP-SGD}}$ are based on [BST14], Algorithm $\mathcal{A}_{\text{obj-pert}}$ is based on [KST12], Algorithm $\mathcal{A}_{\text{FTRL}}$ is

based on [Kai+21b], and Algorithm $\mathcal{A}_{\text{Noise}-\text{FW}(\text{polytope})}$ is based on [TTZ15]. We sincerely apologize for unintentional omission of any related results in this space.

References

[Aba+15] M. Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/ (cit. on p. 222).

[Aba+16] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep Learning with Differential Privacy". In: Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS'16). 2016, pp. 308–318 (cit. on pp. 211, 222).

[AFKT21] H. Asi, V. Feldman, T. Koren, and K. Talwar. "Private Stochastic Convex Optimization: Optimal Rates in L1 Geometry". In: Proceedings of the 38th International Conference on Machine Learning. 2021, pp. 393–403 (cit. on p. 242).

[AS17] N. Agarwal and K. Singh. "The price of differential privacy for online learning". In: International Conference on Machine Learning. PMLR. 2017, pp. 32–40 (cit. on pp. 211, 230).

[BFGT20] R. Bassily, V. Feldman, C. Guzmán, and K. Talwar. "Stability of Stochastic Gradient Descent on Nonsmooth Convex Losses". In: Advances in Neural Information Processing Systems 33 (2020) (cit. on p. 242).

[BFTT19] R. Bassily, V. Feldman, K. Talwar, and A. Thakurta. "Private stochastic convex optimization with optimal rates". In: Advances in Neural Information Processing Systems. 2019, pp. 11279–11288 (cit. on pp. 210, 221).

[Bil08] P. Billingsley. Probability and measure. John Wiley & Sons, 2008 (cit. on pp. 216, 218).

[BLST10] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta. "Discovering frequent patterns in sensitive data". In: KDD. New York, NY, USA, 2010 (cit. on p. 236).

[BM12] M. Bayati and A. Montanari. "The LASSO risk for gaussian matrices". In: IEEE Transactions on Information Theory (2012) (cit. on p. 238).

[BRT09] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. "Simultaneous analysis of Lasso and Dantzig selector". In: The Annals of Statistics (2009), pp. 1705–1732 (cit. on p. 238).

[BST14] R. Bassily, A. Smith, and A. Thakurta. "Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds". In: Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS). 2014, pp. 464–473 (cit. on pp. 211, 215, 221–223, 226, 241, 242).

[Bub15] S. Bubeck. "Convex optimization: Algorithms and complexity". In: Foundations and Trends® in Machine Learning 8.3-4 (2015), pp. 231–357 (cit. on pp. 215, 221, 223, 224).

[Cla10] K. L. Clarkson. "Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm". In: ACM Transations on Algorithms (2010) (cit. on pp. 234, 235).

[CMS11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. "Differentially private empirical risk minimization". In: Journal of Machine Learning Research 12.Mar (2011), pp. 1069–1109 (cit. on pp. 211, 215, 242).

[CSS11] T.-H. H. Chan, E. Shi, and D. Song. "Private and Continual Release of Statistics". In: ACM Trans. on Information Systems Security 14.3 (Nov. 2011), 26:1–26:24 (cit. on pp. 229, 230).

[CT05] E. Candes and T. Tao. "Decoding by linear programming". In: IEEE Transactions on Information Theory 51 (2005) (cit. on p. 238).

[CT07] E. Candes and T. Tao. "The Dantzig selector: Statistical estimation when p is much larger than n". In: The Annals of Statistics (2007), pp. 2313–2351 (cit. on p. 238).

[DJ04] D. Donoho and J. Jin. "Higher criticism for detecting sparse heterogeneous mixtures". In: Annals of Statistics (2004), pp. 962–994 (cit. on p. 238).

[DNPR10] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. "Differential Privacy Under Continual Observation". In: Proc. of the Forty-Second ACM Symp. on Theory of Computing (STOC'10). 2010, pp. 715–724 (cit. on pp. 229, 230).

[Don06] D. L. Donoho. "Compressed sensing". In: IEEE Transactions on Information Theory (2006) (cit. on p. 238).

[DR+14] C. Dwork, A. Roth, et al. "The algorithmic foundations of differential privacy." In: Foundations and Trends in Theoretical Computer Science 9.3-4 (2014), pp. 211–407 (cit. on p. 211).

[DTTZ14] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. "Analyze gauss: optimal bounds for privacy-preserving principal component analysis". In: Proceedings of the forty-sixth annual ACM symposium on Theory of computing. 2014, pp. 11–20 (cit. on p. 242).

[FMTT18] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta. "Privacy Amplification by Iteration". In: 59th Annual IEEE Symp. on Foundations of Computer Science (FOCS). 2018, pp. 521–532 (cit. on p. 242).

[FW56] M. Frank and P. Wolfe. "An algorithm for quadratic programming". In: Naval research logistics quarterly 3.1-2 (1956), pp. 95–110 (cit. on p. 234).

[Haz19] E. Hazan. "Introduction to online convex optimization". In: arXiv preprint arXiv:1909.05207 (2019) (cit. on pp. 227, 232).

[HK12] E. Hazan and S. Kale. "Projection-free Online Learning". In: ICML. 2012 (cit. on p. 235).

[HK14] E. Hazan and S. Kale. "Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization". In: The Journal of Machine Learning Research 15.1 (2014), pp. 2489–2512 (cit. on p. 227).

[HT10] M. Hardt and K. Talwar. "On the geometry of differential privacy". In: Proceedings of the forty-second ACM symposium on Theory of computing. 2010, pp. 705–714 (cit. on p. 215).

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., 2001 (cit. on p. 238).

[Iye+19] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. "Towards practical differentially private convex optimization". In: 2019 IEEE Symposium on Security and Privacy (SP). 2019 (cit. on p. 221).

[Jag13] M. Jaggi. "Revisiting {Frank-Wolfe}: Projection-free sparse convex optimization". In: ICML. 2013 (cit. on pp. 234, 235, 237).

[Kai+21a] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. "Practical and Private (Deep) Learning without Sampling or Shuffling". In: CoRR abs/2103.00039 (2021). URL: https://arxiv.org/abs/2103.00039 (cit. on p. 233).

[Kai+21b] P. Kairouz, B. Mcmahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. "Practical and Private (Deep) Learning Without Sampling or Shuffling". In: Proceedings of the 38th International Conference on Machine Learning. 2021, pp. 5213–5225 (cit. on pp. 211, 230, 243).

[Kas+08] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith. "What Can We Learn Privately?" In: 49th Annual IEEE Symp. on Foundations of Computer Science (FOCS). 2008, pp. 531–540 (cit. on p. 226).

[KST12] D. Kifer, A. Smith, and A. Thakurta. "Private convex empirical risk minimization and high-dimensional regression". In: Conference on Learning Theory. 2012, pp. 25–1 (cit. on pp. 211, 215, 216, 239, 242).

[LMV21] J. Leake, C. McSwiggen, and N. K. Vishnoi. "Sampling matrices from Harish-Chandra–Itzykson–Zuber densities with applications to Quantum inference and differential privacy". In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. 2021, pp. 1384–1397 (cit. on p. 242).

[LV06] L. Lovász and S. Vempala. "Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization". In: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). IEEE. 2006, pp. 57–68 (cit. on p. 215).

[MT07] F. McSherry and K. Talwar. "Mechanism design via differential privacy". In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). IEEE. 2007, pp. 94–103 (cit. on pp. 211, 235, 236).

[MV21] O. Mangoubi and N. K. Vishnoi. "Sampling from Log-Concave Distributions with Infinity-Distance Guarantees and Applications to Differentially Private Optimization". In: arXiv preprint arXiv:2111.04089 (2021) (cit. on p. 215).

[NRVW20] S. Neel, A. Roth, G. Vietri, and S. Wu. "Oracle efficient private non-convex optimization". In: International Conference on Machine Learning. PMLR. 2020, pp. 7243–7252 (cit. on p. 221).

[RWY09] G. Raskutti, M. J. Wainwright, and B. Yu. "Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls". In: ArXiv e-prints (Oct. 2009). arXiv: 0910 . 2042 [math.ST] (cit. on p. 238).

[SCS13] S. Song, K. Chaudhuri, and A. D. Sarwate. "Stochastic gradient descent with differentially private updates". In: 2013 IEEE Global Conference on Signal and Information Processing. IEEE. 2013, pp. 245–248 (cit. on p. 222).

[Sha+11] S. Shalev-Shwartz et al. "Online learning and online convex optimization". In: Foundations and trends in Machine Learning 4.2 (2011), pp. 107–194 (cit. on pp. 227, 230).

[Sha12] S. Shalev-Shwartz. "Online learning and online convex optimization". In: Foundations and Trends in Machine Learning 4.2 (2012), pp. 107–194 (cit. on p. 232).

[She19] O. Sheffet. "Old techniques in differentially private linear regression". In: Algorithmic Learning Theory. PMLR. 2019, pp. 789–827 (cit. on p. 242).

[SSSS09] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. "Stochastic Convex Optimization". In: COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009. 2009 (cit. on p. 210).

[SSTT21] S. Song, T. Steinke, O. Thakkar, and A. Thakurta. "Evading the Curse of Dimensionality in Unconstrained Private GLMs". In: Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. Vol. 130. 2021, pp. 2638–2646 (cit. on p. 222).

[SSZ10] S. Shalev-Shwartz, N. Srebro, and T. Zhang. "Trading accuracy for sparsity in optimization problems with sparsity constraints". In: SIAM Journal on Optimization (2010) (cit. on p. 235).

[ST13a] A. Smith and A. Thakurta. "(Nearly) optimal algorithms for private online learning in full-information and bandit settings". In: Advances in Neural Information Processing Systems. 2013, pp. 2733–2741 (cit. on pp. 211, 229, 230).

[ST13b] A. Smith and A. Thakurta. "Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso". In: COLT. 2013 (cit. on p. 239).

[STT20] S. Song, O. Thakkar, and A. Thakurta. "Characterizing Private Clipped Gradient Descent on Convex Generalized Linear Problems". In: arXiv preprint arXiv:2006.06783 (2020) (cit. on pp. 226, 242).

[STU17] A. Smith, A. Thakurta, and J. Upadhyay. "Is interaction necessary for distributed private learning?" In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE. 2017, pp. 58–77 (cit. on p. 242).

[Tib+97] R. Tibshirani et al. "The lasso method for variable selection in the Cox model". In: Statistics in medicine 16.4 (1997), pp. 385–395 (cit. on p. 238).

[Tib96] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: Journal of the Royal Statistical Society. Series B (Methodological) (1996) (cit. on pp. 233, 238).

[TTZ14a] K. Talwar, A. Thakurta, and L. Zhang. "Private Empirical Risk Minimization Beyond the Worst Case: The Effect of the Constraint Set Geometry". In: CoRR abs/1411.5417 (2014) (cit. on p. 211).

[TTZ14b] K. Talwar, A. Thakurta, and L. Zhang. "Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry". In: arXiv preprint arXiv:1411.5417 (2014) (cit. on p. 242).

[TTZ15] K. Talwar, A. Thakurta, and L. Zhang. "Nearly optimal private lasso". In: Advances in Neural Information Processing Systems 28 (2015) (cit. on pp. 211, 235, 239, 241, 243).

[Vad17] S. Vadhan. "The complexity of differential privacy". In: Tutorials on the Foundations of Cryptography. Springer, 2017, pp. 347–450 (cit. on p. 239).

[Wai06] M. J. Wainwright. "Sharp Thresholds for High-Dimensional and Noisy Recovery of Sparsity Using ℓ_1-Constrained Quadratic Programs". In: IEEE Transactions on Information Theory. 2006 (cit. on pp. 238, 239).

[Was12] L. Wasserman. "Restricted Isometry Property, Rest In Peace". In: Blog-post (2012). URL: http://normaldeviate.wordpress.com/2012/08/07/rip-rip-restricted-isometry-property-rest-in-peace/ (cit. on p. 239).

[WCX19] D. Wang, C. Chen, and J. Xu. "Differentially Private Empirical Risk Minimization with Non-convex Loss Functions". In: Proceedings of the 36th International Conference on Machine Learning. 2019, pp. 6526–6535 (cit. on p. 242).

[Wu+17] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton. "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics". In: Proceedings of the 2017 ACM International Conference on Management of Data. 2017, pp. 1307–1322 (cit. on p. 242).

[You+21] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov. "Opacus: User-Friendly Differential Privacy Library in PyTorch". In: arXiv preprint arXiv:2109.12298 (2021) (cit. on p. 222).

[Zha13] L. Zhang. "Nearly optimal minimax estimator for high dimensional sparse linear regression". In: Annals of Statistics (2013) (cit. on p. 238).