Projection-Free Bandit Optimization with Privacy Guarantees

Alina Ene,[1] **Huy L. Nguyen,** [2] **Adrian Vladu** [3]
[1] Department of Computer Science, Boston University
[2] Khoury College of Computer and Information Science, Northeastern University
[3] CNRS & IRIF, Université de Paris
aene@bu.edu, hu.nguyen@northeastern.edu, vladu@irif.fr

Abstract

We design differentially private algorithms for the bandit convex optimization problem in the projection-free setting. This setting is important whenever the decision set has a complex geometry, and access to it is done efficiently only through a linear optimization oracle, hence Euclidean projections are unavailable (e.g. matroid polytope, submodular base polytope). This is the first differentially-private algorithm for projection-free bandit optimization, and in fact our bound matches the best known non-private projection-free algorithm and the best known private algorithm, even for the weaker setting when projections are available.

Introduction

Online learning is a fundamental optimization paradigm employed in settings where one needs to make decisions in an uncertain environment. Such methods are essential for a range of practical applications: ad-serving (McMahan et al. 2013), dynamic pricing (Lobel, Leme, and Vladu 2018; Mao, Leme, and Schneider 2018), or recommender systems (Abernethy et al. 2007) are only a few examples. These techniques are highly dependent on access to certain user data, such as search history, list of contacts, etc. which may expose sensitive information about a particular person.

As these tools become ubiquitous on the internet, one can witness a surge in the collection of user data at massive scales. This is a tremendous problem, since by obtaining information about the behavior of algorithms run on these data, adversarial entities may learn potentially sensitive information. This could then be traced to a particular user, even if the users were anonymized to begin with (Dwork, Roth et al. 2014).

To mitigate the threat of diminishing user privacy, one can leverage the power of differential privacy (Dwork et al. 2006), a notion of privacy which ensures that the output of an algorithm is not sensitive to the presence of a particular user's data. Therefore, based on this output, one can not determine whether a user presents one or more given attributes.

Differentially private learning algorithms have been studied in several settings, and a large number of recent works addressed the challenge of designing general optimization primitives with privacy guarantees (Jain, Kothari, and Thakurta 2012; Agarwal and Singh 2017; Bassily, Smith, and Thakurta 2014; Abadi et al. 2016; Wang, Ye, and Xu 2017; Iyengar et al. 2019). In this paper, we further advance this line of research by offering differentially private algorithms for a very general task – the bandit convex optimization problem in the case where the space of decisions that the learning algorithm can make exhibits complex geometry.

Bandit convex optimization is an extremely general framework for online learning, which is motivated by the natural setting where, after making a decision, the algorithm only learns the loss associated with its action, and nothing about other possible decisions it could have made (as opposed to the weaker *full information* model where losses associated to all the possible decisions are revealed). Algorithms for this problem are highly dependent on the geometric properties of the space of decisions – and their performance usually depends on the ability to perform certain projections onto this space (Ben-Tal and Nemirovski 2001; Jaggi 2013). For large scale problems, this requirement may be prohibitive, as decisions may have to satisfy certain constraints (the set of recommendations must be diverse enough, or the set of ads to be displayed satisfy a given budget). Projection-free methods overcome this issue by exploiting the fact that some canonical decision sets often encountered in applications (matroid polytope, submodular base polytope, flow polytope, convex relaxations of low-rank matrices) have efficient linear optimization oracles. One can therefore use these efficient oracles in conjunction with the projection-free method to obtain algorithms that can be deployed for real-world applications.

In this work we bridge the requirements of privacy and efficiency for online learning, building on the works of (Garber and Kretzu 2020; Garber and Hazan 2013b), and obtain the first differentially private algorithm for projection-free bandit optimization. To do so we leverage a generic framework for online convex optimization in the presence of noise, which we then adapt to our specific setting in a modular fashion.

Our Contributions. We give the first differentially private algorithm for the bandit convex optimization problem in the projection-free setting (we defer the definition of (ε, δ)-privacy to Definition 2 and the problem statement to the Preliminaries section). We summarize the regret guarantees of our algorithm in the following theorem and compare it with

the state of the art guarantees in the private and non-private settings. Our main focus is on the dependency on the dimension n of the ambient space, the number T of iterations, and the privacy budget ε. For ease of comparison, we use the $\widetilde{O}$ notation to hide poly-logarithmic factors in n and T, as well as parameters such as the Lipschitz constant of the loss functions. The precise guarantees can be found in Lemma 8 (for $(\varepsilon, 0)$-privacy) and Lemma 11 (for (ε, δ)-privacy).

Theorem 1. *Let $\mathcal{D} \subseteq \mathbb{R}^n$ be a convex domain for which we have access to a linear optimization oracle. Assume that for every $t \geq 1$, f_t is convex and L-Lipschitz. Furthermore suppose that $\max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}} \|x - y\| \leq D$. Then there exists an algorithm* PRIVATEBANDIT *(Algorithm 1) which performs projection-free convex optimization in the bandit setting such that one of the following two properties holds:*

- *the algorithm is $(\varepsilon, 0)$-differentially private and, assuming $L = O(1)$ and $D = O(1)$, achieves an expected regret of*

$$\mathcal{R}_T = \widetilde{O}\left(\frac{T^{3/4}n^{3/2}}{\varepsilon}\right).$$

- *the algorithm is (ε, δ)-differentially private and, assuming $L = O(1)$ and $D = O(1)$, achieves an expected regret of*

$$\mathcal{R}_T = \widetilde{O}\left(\frac{(T^{3/4}n^{1/2} + T^{1/2}n)\log^{O(1)}(1/\delta)}{\varepsilon}\right),$$

whenever $\delta = 1/(n+T)^{O(1)}$.

In the non-private setting, the state of the art regret guarantee for projection-free bandit optimization is $\widetilde{O}(n^{1/2}T^{3/4})$ due to Garber and Kretzu (2020).[1] The regret guarantee of our algorithm matches the guarantee of Garber and Kretzu up to a n/ε factor in the $(\varepsilon, 0)$ regime, and a $1/\varepsilon$ factor in the (ε, δ)-regime, whenever $T \geq n^2$.

Prior works in the private setting require projections to be available. The state of the art guarantees for private bandit optimization with projections are achieved by the work of Thakurta and Smith (2013). Thakurta and Smith focus on $(\varepsilon, 0)$-privacy and obtain a regret bound of $\widetilde{O}(nT^{3/4}/\varepsilon)$. A variant of their algorithm can be used for (ε, δ)-privacy and obtains a regret bound of $\widetilde{O}(\sqrt{n}T^{3/4}/\varepsilon)$. Our algorithm's guarantee matches the best guarantee with projections for (ε, δ)-privacy and is worse by a $\sqrt{n}$ factor for $(\varepsilon, 0)$-privacy. We leave it as an interesting open problem to improve the bound for $(\varepsilon, 0)$-privacy to match the one using projections.

Our Techniques. In the process of obtaining our main result, we develop the common abstraction of noisy mirror descent to capture both online bandit optimization and private optimization (the NOISYOCO framework). This allows us to analyze the impact of the noise introduced to protect privacy on the regret of the online optimization. Once the framework is set up, it only remains to analyze the noise level to ensure the appropriate privacy guarantee and one immediately obtains the corresponding regret bound.

[1]Their paper allows for a tradeoff among parameters. This bound is optimized for the case when $T \gg n$.

However, analyzing the noise is in itself a non-trivial challenge. In the case of (ε, δ)-privacy, we give a strong concentration bound allowing us to match the privacy-regret tradeoff achieved with projections (see Lemmas 9 and 10). By using this concentration bound and ignoring the tail of the distribution, we obtain stronger results under (ε, δ)-differential privacy than the straightforward approach. In contrast, in $(\varepsilon, 0)$-differential privacy, one cannot ignore what happens in the tail of the distribution and understanding the algorithm in that regime seems difficult.

Our algorithms use established techniques from differential privacy, such as the Gaussian and Laplacian mechanisms, and tree based aggregation. However, integrating these techniques with optimization methods requires some degree of care. For example, our (ε, δ)-privacy bound is derived using a matrix concentration inequality which crucially relies on a randomized smoothing technique used for obtaining gradient estimates. This is a key ingredient to obtaining the correct $\widetilde{O}(n^{1/2})$ dependence in dimension in the (ε, δ) regime.

Our algorithms can be viewed as a proof of concept for a general approach to deriving differentially private optimization methods. Previous results in this area can be recovered by following our approach: inject the maximum amount of noise as to not change the convergence rate asymptotically, then analyze the privacy loss. This is very simple, but it paves the way for further development of practical differentially private algorithms, without requiring major changes in their implementation – simply replace certain components of the algorithm with a black box implementation of the required differentially private mechanism.

We believe our framework is general and it facilitates further progress in differentially private optimization. We demonstrate our framework by instantiating it with the Laplace mechanism (to obtain an $(\varepsilon, 0)$-private algorithm) and with the Gaussian mechanism (to obtain an (ε, δ)-private algorithm). It would be interesting to apply our framework to other notions of differential privacy, such as concentrated differential privacy (Dwork and Rothblum 2016; Bun and Steinke 2016) and Renyi differential privacy (Mironov 2017).

Other Related Work. The theory of online convex optimization is truly extensive, and has seen a lot of developments in recent years. Here we only discuss the relevant works that involve projection-free and/or differentially private online learning algorithms. The class of projection-free online learning algorithms was initiated by the work of Hazan and Kale (2012) in the context of online convex optimization, where full gradients are revealed after making a decision. This was further extended to multiple regimes (Garber and Hazan 2013b,a, 2015) including the bandit setting (Chen, Zhang, and Karbasi 2018; Garber and Kretzu 2020).

As discussed above, Thakurta and Smith (2013) achieve the state of the art regret guarantee for $(\varepsilon, 0)$-private online bandit optimization when projections are available. For general Lipschitz functions, their regret is $\widetilde{O}(nT^{3/4}/\varepsilon)$. In the specific case where the adversary is oblivious and the

loss functions are strongly-convex, they improve this to $\widetilde{O}(nT^{2/3}/\varepsilon)$.

In a different line of work, Agarwal and Singh (2017) obtained improved bounds for the case where losses are linear functions and for the multi-armed bandit problem (a generalization of the learning with experts framework), with regret $\widetilde{O}(T^{2/3}/\varepsilon)$ and $\widetilde{O}(nT^{2/3}/\varepsilon)$ respectively. These results, however, concern only a restricted class of bandit optimization problems, so for the general bandit convex optimization problem the result of Thakurta and Smith (2013) still stands as the best one.

In fact, even in the non-private setting, improving the regret of the bandit convex optimization problem from $\widetilde{O}(T^{3/4})$ to $\widetilde{O}(T^{2/3})$ (Awerbuch and Kleinberg 2004; Dani and Hayes 2006) or below (Dani, Kakade, and Hayes 2008; Abernethy, Hazan, and Rakhlin 2008; Bubeck, Lee, and Eldan 2017) requires stronger access to the set of actions than just projections (such as via a self-concordant barrier), and involves performing expensive computations. Indeed, Bubeck, Lee, and Eldan (2017) are the first to achieve both optimal regret $\widetilde{O}(T^{1/2})$ and polynomial running time per iteration.

Preliminaries

Bandit Convex Optimization. In the bandit convex optimization problem, an algorithm iteratively selects actions $\boldsymbol{x}_t$ (using a possibly randomized strategy) from a convex set $\mathcal{D} \subseteq \mathbb{R}^n$. After selecting an action, the loss caused by this choice $f_t(\boldsymbol{x}_t)$ is revealed, where $f_t : \mathcal{D} \to \mathbb{R}$ is a convex function unknown to the algorithm.

After T iterations, the algorithm compares its total loss $\sum_{t=1}^{T} f_t(\boldsymbol{x}_t)$ to the smallest loss it could have incurred by choosing a fixed strategy throughout all the iterations $\min_{\boldsymbol{x}\in\mathcal{D}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$. The difference between these two losses is called *regret*:

$$\mathcal{R}_T = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x}\in\mathcal{D}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

and the goal is to minimize its expectation over the randomized choices of the algorithm.

Differential Privacy. Differential privacy (Dwork et al. 2006) is a rigorous framework used to control the amount of information leaked when performing computation on a private data set. In our framework, we seek algorithms which ensure that the amount of information an adversary can learn about a particular loss function f_t is minimal, i.e. it is almost independent on whether f_t appears or not in the sequence of loss functions occurring throughout the execution of the algorithm. For completeness, we define differential privacy in the context of loss functions encountered in the bandit convex optimization problem.

Definition 2 ((ε, δ)-differential privacy)**.** A randomized online learning algorithm $\mathcal{A}$ on the action set $\mathcal{D}$ is (ε, δ)-differentially private if for any two sequences of loss functions $F = (f_1, \ldots, f_T)$ and $F' = (f'_1, \ldots, f'_T)$ differing in at most one element, for all $S \subseteq \mathcal{D}^T$ it holds that

$$\Pr[\mathcal{A}(F) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(F') \in S] + \delta\,.$$

One obstacle that may occur in the context of bandit optimization is that changing a single loss function may alter the set of actions returned in the future by the algorithm.

The Projection-Free Setting. While classical online optimization methods have a long history of developments, these rely in general on the ability to perform projections onto the feasible set $\mathcal{D}$ of actions. For example, one may want to choose actions that correspond to points inside a matroid polytope, or other complicated domains. In such situations, it is computationally infeasible to perform projections onto $\mathcal{D}$, and designing algorithms where all the actions lie inside this domain becomes a challenging task. In the case of online optimization, this issue is mitigated by *projection-free methods* (Jaggi 2013; Garber and Hazan 2015; Dudik, Harchaoui, and Malick 2012; Shalev-Shwartz, Gonen, and Shamir 2011), where the complexity of the high complexity of the description of $\mathcal{D}$ is balanced by the existence of a linear optimization oracle over this domain. Among these, the conditional gradient method (also known as Frank-Wolfe) (Bubeck et al. 2015) is the best known one.

In our setting, we treat the case where, although $\mathcal{D}$ may be very complicated, we have access to a linear optimization oracle that, given any direction $\boldsymbol{v} \in \mathbb{R}^n$, returns $\arg\min_{\boldsymbol{x}\in\mathcal{D}} \langle \boldsymbol{v}, \boldsymbol{x} \rangle$. Such oracles are easily available for interesting domains such as the matroid polytope or the submodular base polytope.

Parameters and Assumptions. We write vectors and matrices in bold. We use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to represent inner products, and $\|\boldsymbol{x}\|$ to represent the ℓ_2 norm of a vector $\|\boldsymbol{x}\| = \left(\sum_i x_i\right)^{1/2}$. When considering other norms than ℓ_2 we explicitly specify them $\|\boldsymbol{x}\|_p = \left(\sum_i x_i^p\right)^{1/p}$. We let B_p^n be the n dimensional unit ℓ_p ball and S_p^n the boundary of B_p^n i.e. the n dimensional unit ℓ_p sphere. We consider optimizing over a convex domain $\mathcal{D} \subseteq \mathbb{R}^n$, for which we have access to a linear optimization oracle. We define the diameter of the domain as $D = \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}} \|x - y\|$. We say that a function $f : \mathcal{D} \to \mathbb{R}$ is L-Lipschitz iff $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$ and that a differentiable function f is β-smooth iff $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq \beta\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$. We say that f is α-strongly convex iff $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \geq \alpha\|\boldsymbol{x} - \boldsymbol{y}\|$. In our setting, all functions f_t satisfy the standard assumption of being L-Lipschitz.

Just like in prior works (Thakurta and Smith 2013; Agarwal and Singh 2017), we further assume that the number of iterations T we run the algorithm for is known ahead of time. This assumption can be eliminated via a standard reduction using the doubling trick (see (Auer et al. 1995) and (Chen, Zhang, and Karbasi 2018)), which invokes the base algorithm repeatedly by doubling the horizon T at each invocation, at the expense of adding an extra $O(\log T)$ factor in the privacy loss.

We further assume that all f_t's are defined within a region that slightly exceeds the boundary of $\mathcal{D}$. This assumption is required, since one of the techniques employed here requires having f_t defined over $\mathcal{D} \oplus \zeta B_2^n$ for a small scalar ζ. This assumption can be removed via a simple scaling trick, whenever $\mathcal{D}$ contains an ℓ_2 ball centered at the origin (similarly

to (Garber and Kretzu 2020)); we explain how to do so in the full version (Ene, Nguyen, and Vladu 2021).

Finally, in order to be able to appropriately privatize the losses $f_t(\boldsymbol{x}_t)$, we need to bound their magnitude. To this end, we assume that each f_t achieves 0 loss at some point within $\mathcal{D}$, which via the Lipschitz condition and the diameter bound automatically implies that $|f_t(\boldsymbol{x}_t)| \leq LD$ for all t. Other related works (Flaxman, Kalai, and McMahan 2004; Agarwal, Dekel, and Xiao 2010; Thakurta and Smith 2013) simply use a fixed upper bound $|f_t(\boldsymbol{x}_t)| \leq B$ for some fixed parameter B, but we prefer this new convention to reduce the number of parameters to control, since we are focused mainly in the regret dependence in T, n and ε.

Mirror Maps and the Fenchel Conjugate. In general, convex optimization implicitly relies on the existence of a *mirror map* $\omega : \mathcal{D} \to \mathbb{R}$ with desirable properties (see (Ben-Tal and Nemirovski 2001) for an extensive treatment of these objects). This is used in order to properly interface iterates and gradient updates, since in Banach spaces these are of different types. In this paper, we use $\omega(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, although other choices can be used depending on the geometry of $\mathcal{D}$. We define the Fenchel conjugate of ω as $\omega^* : \mathbb{R}^n \to \mathbb{R}$ such that

$$\omega^*(\boldsymbol{y}) = \max_{\boldsymbol{x} \in \mathcal{D}} \langle \boldsymbol{y}, \boldsymbol{x} \rangle - \omega(\boldsymbol{x}) \,. \tag{0.1}$$

Furthermore, if ω is strongly convex, then ω^* is smooth and differentiable (Nesterov 2005), and its gradient satisfies

$$\nabla \omega^*(\boldsymbol{y}) = \arg\max_{\boldsymbol{x} \in \mathcal{D}} \langle \boldsymbol{y}, \boldsymbol{x} \rangle - \omega(\boldsymbol{x}) \,. \tag{0.2}$$

Smoothing. We use the randomized smoothing technique from (Flaxman, Kalai, and McMahan 2004) in order to smoothen the loss functions f_t. This technique is crucial to obtain gradient estimators despite having only value access.

Lemma 3 ((Flaxman, Kalai, and McMahan 2004))**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex and L-Lipschitz function. Then the smoothing $\widehat{f}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u} \sim B_2^n} f(\boldsymbol{x} + \zeta \boldsymbol{u})$ satisfies the following properties: (1) $\left| f(\boldsymbol{x}) - \widehat{f}(\boldsymbol{x}) \right| \leq \zeta L$, (2) $\widehat{f}$ is convex and L-Lipschitz, (3) $\nabla \widehat{f}(\boldsymbol{x}) = \frac{n}{\zeta} \cdot \mathbb{E}_{\boldsymbol{u} \sim S_2^n} f(\boldsymbol{x} + \zeta \boldsymbol{u}) \cdot \boldsymbol{u}$.*

Tree Based Aggregation. An essential ingredient of the algorithm is maintaining partial sums of the gradient estimators witnessed so far. We use a variant of the algorithm from (Dwork et al. 2010; Jain, Kothari, and Thakurta 2012), as implemented in (Agarwal and Singh 2017). We use the algorithm as a black box and only rely on its properties that are stated in Theorem 4 below. We include a description of the TREEBASEDAGG algorithm in the full version (Ene, Nguyen, and Vladu 2021) for completeness.

Theorem 4 ((Jain, Kothari, and Thakurta 2012; Agarwal and Singh 2017))**.** *Let $\{\ell_t\}_{t=1}^T$ be a sequence of vectors in $\mathbb{R}^n$, and let Y_1 and Y_2 be promises such that $\|\ell_t\|_1 \leq Y_1$ and $\|\ell_t\|_2 \leq Y_2$ for all t. Let $\varepsilon, \delta > 0$, and $\lambda \geq \frac{Y_1 \log T}{\varepsilon}$ and $\sigma \geq \frac{Y_2}{\varepsilon} \log T \log \frac{\log T}{\delta}$.*

There is an algorithm, TREEBASEDAGG*, that first outputs $\widehat{L}_0$ and then iteratively takes ℓ_t as input and returns an approximate partial sum $\widehat{L}_t$ for $1 \leq t \leq T$. The algorithm can be specified with a noise distribution $\mathcal{P}$ over $\mathbb{R}^n$ so that the output sequence $\{\widehat{L}_t\}_{t=1}^T$ satisfies $\widehat{L}_t = \sum_{s=1}^t \ell_s + \sum_{r=1}^{\lceil \log T \rceil} Z_r$, where $Z_r \sim \mathcal{P}$, and furthermore:*

- *when $\mathcal{P}$ is coordinate-wise $Lap(0, \lambda)$, the sequence $\{\widehat{L}_t\}_{t=1}^T$ is $(\varepsilon, 0)$-differentially private.*
- *when $\mathcal{P}$ is coordinate-wise $\mathcal{N}(0, \sigma^2)$, the sequence $\{\widehat{L}_t\}_{t=1}^T$ is (ε, δ)-differentially private.*

Algorithm

The algorithm is described in Algorithm 1. It builds on the work of Garber and Kretzu (2020) and uses the smoothing (Lemma 3) and tree aggregation (Theorem 4) routines designed in previous work (see the preliminaries section). The algorithm follows the structure of an online mirror descent algorithm. It performs a sequence of iterations, and in each iteration it makes a guess x_t based on the previous outcomes. The iterations are divided into T_{round} batches, each of size T_{batch} (thus, $T = T_{\text{round}} \cdot T_{\text{batch}}$). Each batch R is treated as a round for online mirror descent with a twist: in parallel, we compute the best regularized response $\widetilde{\boldsymbol{x}}_R$ for the revealed outcomes in the first $R-1$ batches (line 14) and use the previously computed $\widetilde{\boldsymbol{x}}_{R-1}$ for all iterations in batch R (lines 6 to 12). Three notices are in order:

- Computing the best regularized response to previous outcomes requires maintaining the sum of gradients in previous rounds. The tree-based aggregation method (Theorem 4) is used to maintain these sums accurately while preserving privacy (line 13).
- In each iteration of a batch, the algorithm only has access to the function value, not the gradient, so we use the smoothing technique (Lemma 3): the function value at a perturbation of $\widetilde{\boldsymbol{x}}_{R-1}$ is used to obtain a stochastic estimate of the gradient of a smoothed proxy of the objective function. Thus each iteration in the same batch uses a different perturbation of the same response $\widetilde{\boldsymbol{x}}_{R-1}$.
- We only compute an approximation of the best regularized response, using the conditional gradient method in line 14. The precision to which this is computed is chosen in such a way that the number of iterations required by conditional gradient matches the number of iterations in a batch, so that we can charge each call to the linear optimization oracle over $\mathcal{D}$ to one iteration of the bandit algorithm.

The algorithm needs to be specified with a noise distribution $\mathcal{P}$ over $\mathbb{R}^n$, which we use for privatizing the partial sums in order to strike the right tradeoff between privacy and regret. To obtain an $(\varepsilon, 0)$-private algorithm, we set $\mathcal{P}$ to be coordinate-wise Laplace noise $Lap(0, \lambda)$. To obtain an (ε, δ)-private algorithm, we set $\mathcal{P}$ to be coordinate-wise Gaussian noise $\mathcal{N}(0, \sigma^2)$. The precise choice for the parameters λ and σ^2 are established in Lemmas 8 and 11.

Our algorithm can be viewed as a slight simplification and generalization of the algorithm of Garber and Kretzu (2020). Specifically, we do not require any specific stopping conditions and case analysis for solving the inner conditional gradient routine and we can extend it to more general geometries defined by the mirror map. Additionally, the

Algorithm 1 PRIVATEBANDIT$(T, \mathcal{P}, D)$

input time horizon T, symmetric noise distribution $\mathcal{P}$, diameter of domain D.

$T_{\text{round}} = T^{1/2}, T_{\text{batch}} = \frac{T}{T_{\text{round}}}, \eta = \frac{D}{T^{3/4} n^{1/2} L}, \zeta = \frac{D n^{1/2}}{T^{1/4}}$.
Initialize TREEBASEDAGG for a sequence of length T_{round} and noise $\mathcal{P}$.
for $R = 1$ **to** T_{round} **do**
  execute in parallel:
  $\widetilde{\boldsymbol{g}}_R = 0$
  for $r = 1$ **to** T_{batch} **do**
    $t = (R-1)T_{\text{batch}} + r$
    Sample $\boldsymbol{u}_t \sim S_2(1)$
    $\boldsymbol{x}_t = \widetilde{\boldsymbol{x}}_{R-1} + \zeta \boldsymbol{u}_t$
    Query $F_t = \frac{n}{\zeta} f_t(\widetilde{\boldsymbol{x}}_{R-1} + \zeta \boldsymbol{u}_t)$
    $\widetilde{\boldsymbol{g}}_R = \widetilde{\boldsymbol{g}}_R + F_t \cdot \boldsymbol{u}_t$
  end for
  // Update the partial sum of noisy gradients:
  $\widetilde{\boldsymbol{s}}_R = \text{TREEBASEDAGG}\,(\widetilde{\boldsymbol{g}}_R, R)$
  Solve via conditional gradient

$$\min_{\boldsymbol{x} \in \mathcal{D}} \frac{1}{2} \|\boldsymbol{x}\|_2^2 - \langle \eta \widetilde{\boldsymbol{s}}_{R-1}, \boldsymbol{x} \rangle$$

  to precision $\varepsilon_{\text{cg}} = D / T_{\text{batch}}^{1/2}$. Let $\widetilde{\boldsymbol{x}}_R$ be the output.
end for

NOISYOCO framework allows us to handle the noise introduced by the differentially private mechanisms without making any further changes to the algorithm or its analysis. This framework may be of further interest for designing differentially private optimization methods.

In the following section, we show the following regret guarantee:

Lemma 5. *Let $\mu = \mathbb{E}_{X \sim \mathcal{P}} \|X\|$, and let D be the diameter of the domain $\mathcal{D}$, n the ambient dimension, and L an upper bound on the Lipschitz constant of the loss functions f_t. Algorithm* PRIVATEBANDIT *achieves a regret of*

$$O\left(T^{3/4} n^{1/2} L D + T^{1/4} D \mu \log T\right).$$

Noisy Mirror Descent Framework and Regret Analysis

In this section, we sketch the regret analysis for Algorithm 1. We derive the algorithm's regret guarantee via the NOISYOCO framework, a meta-algorithm for online convex optimization with noise. We show that Algorithm 1 is an instantiation of this meta-algorithm and we derive its regret guarantees from the guarantee for NOISYOCO.

Noisy Mirror Descent Framework

Here we describe and analyze the NOISYOCO algorithm (Algorithm 2) for online convex optimization with noise. We assume that we perform online convex optimization over a convex domain $\mathcal{D}$ endowed with a strongly convex mirror map $\omega : \mathcal{D} \to \mathbb{R}$ such that $\max_{\boldsymbol{x} \in \mathcal{D}} \omega(\boldsymbol{x}) \le D_\omega^2$. We also assume (κ, γ)-noisy gradient access, defined as follows:

- a noisy gradient oracle for f_t; given $\boldsymbol{x}$, it returns a randomized $\widetilde{\boldsymbol{g}} = \text{NOISYGRAD}(f_t, \boldsymbol{x})$ such that $\mathbb{E}\widetilde{\boldsymbol{g}} = \nabla f_t(\boldsymbol{x})$, and $\mathbb{E}\|\widetilde{\boldsymbol{g}}\|^2 \le \kappa^2$,
- a noisy gradient oracle for ω^*; given $\boldsymbol{g}$, it returns a randomized $\widetilde{\boldsymbol{x}} = \text{NOISYMAP}(\boldsymbol{g})$ such that $\mathbb{E}\|\nabla \omega^*(\boldsymbol{g}) - \widetilde{\boldsymbol{x}}\| \le \gamma$.

Algorithm 2 NOISYOCO(T)

input time horizon T.

$\eta = D_\omega^{1/2} / (\kappa T^{1/2}), \boldsymbol{s}_0 = 0$
for $t = 1$ **to** T **do**
  $\widetilde{\boldsymbol{x}}_t = \text{NOISYMAP}(-\eta \boldsymbol{s}_{t-1})$
  output $\widetilde{\boldsymbol{x}}_t$ and **query** $\widetilde{\boldsymbol{g}}_t = \text{NOISYGRAD}(f_t, \widetilde{\boldsymbol{x}}_t)$
  $\boldsymbol{s}_t = \boldsymbol{s}_t + \widetilde{\boldsymbol{g}}_t$
end for

Under these conditions we can derive the following regret guarantee. We give the proof in the full version (Ene, Nguyen, and Vladu 2021).

Lemma 6. *Given an instance of online convex optimization with (κ, γ)-noisy gradient access, the algorithm* NOISYOCO *obtains an expected regret of*

$$\mathcal{R}_T = O\left(T^{1/2} \kappa D_\omega + T \kappa \gamma\right).$$

Regret Analysis

The regret analysis for Algorithm 1 is based on the guarantee for NOISYOCO from Lemma 6. It follows from mapping the steps in Algorithm 1 to the framework from NOISYOCO, and bounding the parameters involved. In order to do so, we explain how the NOISYGRAD and NOISYMAP routines are implemented by Algorithm 1. We then proceed to bound the κ and γ parameters corresponding to this specific instantiation, which will yield the desired result. Here we describe the steps required for analysis. We give the detailed proofs in the full version (Ene, Nguyen, and Vladu 2021).

The first step is to reduce the problem to minimizing regret on a family of functions $\{\widetilde{f}_R\}_{R=1}^{T_{\text{round}}}$, where $\widetilde{f}_R = \sum_{t=1}^{T_{\text{round}}} \widehat{f}_{(T_{\text{batch}}-1) \cdot R + t}$. This introduces two sources of error: one from using the smoothed $\widehat{f}$ instead of f, and another from using different iterates $x_t = \widetilde{\boldsymbol{x}}_{R-1} + \boldsymbol{u}_t$ in the same round, even though batching iterations effectively constrains all the iterates in a fixed batch to be equal. These errors are easy to control, and add at most $O(T\zeta L)$ in regret.

For the family of functions $\{\widetilde{f}_R\}_{R=1}^{T_{\text{round}}}$ we implement NOISYGRAD as:

$$\begin{aligned} \widetilde{\boldsymbol{g}}_R &:= \text{NOISYGRAD}(\widetilde{f}_R, \boldsymbol{x}_t) \\ &= \sum_{t=(R-1)T_{\text{batch}}+1}^{RT_{\text{batch}}} \frac{n}{\zeta} f_t(\boldsymbol{x}_t + \zeta \boldsymbol{u}_t) \boldsymbol{u}_t \,, \end{aligned}$$

which is an unbiased estimator for $\nabla \widehat{f}_R(\boldsymbol{x}_t)$, per Lemma 3. Thus we bound κ^2 by showing that $\mathbb{E}\|\widetilde{\boldsymbol{g}}_R\|^2 \le T_{\text{batch}} \cdot (LDn/\zeta)^2 + T_{\text{batch}}^2 L^2$.

Furthermore, the output of NOISYMAP is implemented in line 14 by running conditional gradient to approximately minimize a quadratic over the feasible domain $\mathcal{D}$. The error in the noisy map implementation comes from (1) only approximately minimizing the quadratic, (2) using a noisy partial sum of gradient estimators rather than an exact one, and (3) using a stale partial sum approximation $\widetilde{\boldsymbol{s}}_{R-1}$ for round $R+1$, instead of $\widetilde{\boldsymbol{s}}_R$. We show that the error parameter corresponding to this NOISYMAP implementation can be bounded as $\gamma \leq \eta\left(\lceil \log T \rceil \cdot \mu + \kappa\right) + \sqrt{20}\frac{D}{\sqrt{T_{\text{batch}}}}$.

In the full version (Ene, Nguyen, and Vladu 2021), we bound the specific parameters corresponding to these implementations. Plugging these with bounds into Lemma 6 yields the proof of Lemma 5, after appropriately balancing the parameters $T_{\text{batch}}, T_{\text{round}}, \zeta$.

Privacy Analysis

In this section, we instantiate Algorithm 1 with appropriate noise distribution $\mathcal{P}$ in order to derive our $(\varepsilon, 0)$-private algorithm and our (ε, δ)-private algorithm. As mentioned earlier, we use Laplace noise for $(\varepsilon, 0)$-privacy and obtain the guarantee in Lemma 8, and we use Gaussian noise for (ε, δ)-privacy and obtain the guarantee in Lemma 11.

First, we describe the proofs for the $(\varepsilon, 0)$-privacy regime, where we employ Laplace noise, since they show how this framework allows us to trade regret and privacy.

Lemma 7 (Privacy with Laplace noise)**.** *Let $\mathcal{P}$ be coordinate-wise $Lap(0, T^{1/2}nL\log T/\varepsilon)$. The algorithm* PRIVATEBANDIT$(T, \mathcal{P})$ *is $(\varepsilon, 0)$-differentially private.*

Proof. First we bound the ℓ_1 norm of the vectors whose partial sums are maintained by the tree based aggregation method in Algorithm 1. Since each vector contributing to that partial sum is obtained by adding up $T_{\text{batch}} = T^{1/2}$ vectors, each of which is a unit ℓ_2 vector scaled by a constant that is absolutely bounded by $M = LD\frac{n}{\zeta} = T^{1/4}n^{1/2}L$, we naively bound the ℓ_1 norm of each of them by $T_{\text{batch}} \cdot n^{1/2} \cdot M \leq T^{1/2}nL$.

Therefore, by Theorem 4, releasing $T_{\text{round}} = T^{1/2}$ such partial sums causes a loss of privacy of at most ε whenever

$$\lambda \geq \frac{T^{1/2}nL\log T}{\varepsilon}\,.$$

□

Using Lemma 7 we can now bound the regret of the $(\varepsilon, 0)$-differentially private algorithm.

Lemma 8 (Regret with Laplace noise)**.** *Let $\mathcal{P}$ be coordinate-wise $Lap(0, T^{1/2}nL\log T/\varepsilon)$. The algorithm* PRIVATEBANDIT$(T, \mathcal{P})$ *has regret*

$$\mathcal{R}_T = O\left(T^{3/4}n^{1/2}LD + \frac{T^{3/4}n^{3/2}LD\log^2 T}{\varepsilon}\right).$$

Proof. Per Lemma 5 we only need to upper bound the expected ℓ_2 norm of an n-dimensional vector where each coordinate is independently sampled from $Lap(0, T^{1/2}nL\log T/\varepsilon)$. Indeed, we have

$$\mu = O\left(n^{1/2} \cdot T^{1/2}nL\log T/\varepsilon\right).$$

Plugging this into the regret guarantee from Lemma 5 gives the desired result. □

We note that the regret guarantee we achieved has an undesirable dependence in dimension. In the remainder of this section, we show that we can obtain improved guarantees if we settle for (ε, δ)-differential privacy instead, which we achieve by using Gaussian noise. This is also more properly suited to our setting, since the regret bound we proved depends on ℓ_2 norms of the injected noise vectors, which is exactly what governs the privacy loss in this case. A novel and precise error analysis in Lemmas 9 and 10 enables us to obtain the same regret bound as when projection is available for (ε, δ)-privacy.

We additionally use the fact that the randomized smoothing technique further constrains the norm of the vectors $\widetilde{\boldsymbol{g}}_R$ we employ to update the partial sums, with high probability. In order to do this, we use a concentration inequality for random vectors (Lemma 9) which allows us to obtain an improved guarantee on privacy, at the expense of a slight increase in the δ parameter. Compared to the $(\varepsilon, 0)$ case, allowing this low probability failure event enables us to save roughly a factor of $\widetilde{O}(T^{1/4}n^{-1/2})$ in the norm of the vectors we use to update the partial sums via tree based aggregation. In turn, this allows us to use less noise to ensure privacy, and therefore we obtain an improved regret.

We start with the following lemma, which establishes a high probability bound on the ℓ_2 norm of a sum of random vectors multiplied by an adversarially chosen set of scalars.

Lemma 9. *Let $\boldsymbol{u}_1, \cdots \boldsymbol{u}_k \sim B_2(1)$ be a set of independent random vectors in $\mathbb{R}^n$. Then, with probability at least $1-\delta$, one has that for any vector $\boldsymbol{c} \in \mathbb{R}^k$ such that $\|\boldsymbol{c}\| \leq \Delta$:*

$$\left\|\sum_{i=1}^k \boldsymbol{u}_i c_i\right\| \leq 10\Delta\left(\log\frac{n+k}{\delta} + \sqrt{\left(1+\frac{k}{n}\right)\log\frac{n+k}{\delta}}\right)$$

Proof. Consider the family of matrices $\{\boldsymbol{Z}_i\}_{i=1}^k \in \mathbb{R}^{n\times k}$ where $\boldsymbol{Z}_i$ has its i^{th} column equal to $\boldsymbol{u}_i$ and all the other entries are 0. Therefore $\mathbb{E}\boldsymbol{Z}_k = 0$ and $\|\boldsymbol{Z}_k\| \leq 1$. Furthermore, by definition $\boldsymbol{Z}_i\boldsymbol{Z}_i^\top = \boldsymbol{u}_i\boldsymbol{u}_i^\top$ and $\boldsymbol{Z}_i^\top\boldsymbol{Z}_i = \|\boldsymbol{u}_i\|^2 \cdot \boldsymbol{1}_i\boldsymbol{1}_i^\top$. Therefore $\mathbb{E}\boldsymbol{Z}_i\boldsymbol{Z}_i^\top = \boldsymbol{I}/n$ and $\mathbb{E}\boldsymbol{Z}_i\boldsymbol{Z}_i^\top = \boldsymbol{1}_i\boldsymbol{1}_i^\top$.

So

$$\begin{aligned}\sigma^2 &= \max\left\{\left\|\mathbb{E}\sum_{i=1}^k \boldsymbol{Z}_i\boldsymbol{Z}_i^\top\right\|, \left\|\mathbb{E}\sum_{i=1}^k \boldsymbol{Z}_i^\top\boldsymbol{Z}_i\right\|\right\} \\ &= \max\left\{\left\|\frac{k}{n}\cdot\boldsymbol{I}\right\|, \|\boldsymbol{I}\|\right\} \leq 1 + k/n\,.\end{aligned}$$

Letting $\boldsymbol{Z} = \sum_{i=1}^k \boldsymbol{Z}_i$, and using matrix Bernstein (Tropp et al. 2015), we see that

$$\Pr\left[\|\boldsymbol{Z}\| \geq t\right] \leq (n+k)\exp(-t^2/(2(1+k/n)+2t/3))\,.$$

Hence for $t = 10\left(\log\frac{n+k}{\delta} + \sqrt{\left(1+\frac{k}{n}\right)\log\frac{n+k}{\delta}}\right)$ one has that $\|\boldsymbol{Z}\| \le t$ with probability at least $1-\delta$.

Therefore with probability at least $1-\delta$ we have $\|\boldsymbol{Z}\boldsymbol{c}\| \le \|\boldsymbol{Z}\|\|\boldsymbol{c}\| \le 10\Delta\left(\log\frac{n+k}{\delta} + \sqrt{\left(1+\frac{k}{n}\right)\log\frac{n+k}{\delta}}\right)$ which implies what we needed. □

Using the above lemma, we can obtain a tighter bound the privacy loss when using Gaussian noise.

Lemma 10 (Privacy with Gaussian noise)**.** *Let $\mathcal{P}$ be coordinate-wise $\mathcal{N}(0,\sigma^2)$, where*

$$\sigma = \frac{T^{1/4}n^{1/2}L\log T\log(T/\delta)}{\varepsilon} \cdot \left(\log\frac{n+T}{\delta} + \sqrt{\left(1+\frac{T^{1/2}}{n}\right)\log\frac{n+T}{\delta}}\right)$$

The algorithm PRIVATEBANDIT$(T,\mathcal{P})$ *is (ε,δ)-differentially private.*

Proof. Using Lemma 9 and union bound we have that with probability at least $1-\delta_0 T^{1/2}$ the ℓ_2 norm of each of the T_{batch} vectors contributing to the partial sums maintained in the tree based aggregation routine is $M = O\left(T^{1/4}n^{1/2}L\left(\log\frac{n+T}{\delta_0} + \sqrt{(1+\frac{T_{\text{batch}}}{n})\log\frac{n+T}{\delta_0}}\right)\right)$.

By Theorem 4 maintaining these partial sums is thus $(\varepsilon, \delta_0 T^{1/2}+\delta_1)$-differentially private, where $\varepsilon = \frac{M\log T\log(T/\delta_1)}{\sigma}$. Hence setting $\delta_1 = \delta/2$, $\delta_0 = \delta/(2T^{1/2})$ and

$$\sigma = \frac{M\log T\log(T/\delta_1)}{\varepsilon} = \frac{T^{1/4}n^{1/2}L\log T\log(T/\delta)}{\varepsilon} \cdot \left(\log\frac{n+T}{\delta} + \sqrt{\left(1+\frac{T^{1/2}}{n}\right)\log\frac{n+T}{\delta}}\right)$$

yields an (ε,δ)-differentially private algorithm. □

Lemma 11 (Regret with Gaussian noise)**.** *Let $\delta = 1/(n+T)^{O(1)}$. The algorithm* PRIVATEBANDIT$(T,\mathcal{N}(0,\sigma^2))$ *where σ is chosen according to Lemma 10 such that the algorithm is (ε,δ)-private has regret*

$$\mathcal{R}_T = O\Bigg(T^{3/4}n^{1/2}LD + \frac{T^{1/2}nLD\log^2 T\log(T/\delta)\log((n+T)/\delta)}{\varepsilon} + \frac{T^{3/4}n^{1/2}LD\log^2 T\log(T/\delta)\sqrt{\log((n+T)/\delta)}}{\varepsilon}\Bigg).$$

Proof. Per Lemma 5 we only need to upper bound the expected norm of an n-dimensional vector where each coordinate is sampled from $\mathcal{N}(0,\sigma^2)$. In this case we have $\mu = O(n^{1/2}\sigma)$, so plugging it into the regret guarantee from Lemma 5 we obtain regret

$$\mathcal{R}_T = O\left(T^{3/4}n^{1/2}LD + T^{1/4}n^{1/2}D\sigma\log T\right)$$

which implies the result after substituting σ. □

The proof of Theorem 1 now follows from combining Lemmas 7, 8, 10 and 11.

Discussion and Open Problems

We saw how one can derive differentially private algorithm starting from a very basic framework for noisy online convex optimization. Our analysis builds on advances in both differential privacy and online convex optimization, combines their techniques in non-trivial ways and introduces new ideas. Among others, a novel and precise error analysis in Lemmas 9 and 10 enables us to obtain the same regret bound as when projection is available for (ε,δ)-privacy, in contrast to $(\varepsilon,0)$-privacy. To the best of our knowledge, this is a rare case where such a difference between the two privacy settings arise. We think it is an interesting direction for future work to obtain an analogous improvement even in the $(\varepsilon,0)$-privacy setting.

It would be interesting to see if this generic method in conjunction with tools from differential privacy can be used to obtain more private learning algorithms. A few outstanding questions remain. Since $\widetilde{O}(T^{3/4})$ is also the best known regret bound in the non-private setting, it would be interesting to improve this result, which may lead to an improved differentially private algorithm. Furthermore, in the non-private setting with projections, obtaining algorithms with lower regret requires a stronger control of the geometry of the domain. This makes differential privacy more difficult to achieve since even the simplest algorithms with improved regret require solving a linear system of equations, which is much more sensitive to noise than vanilla gradient steps. Developing a more fine-grained understanding of these problems via differential privacy poses an exciting challenge.

Acknowledgments

AE was supported in part by NSF CAREER grant CCF-1750333, NSF grant CCF-1718342, and NSF grant III-1908510. HLN was supported in part by NSF CAREER grant CCF-1750716 and NSF grant CCF-1909314. AV was supported in part by NSF grant CCF-1718342.

References

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

Abernethy, J.; Canini, K.; Langford, J.; and Simma, A. 2007. Online collaborative filtering. Technical report, University of California at Berkeley.

Abernethy, J. D.; Hazan, E.; and Rakhlin, A. 2008. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory (COLT)*, 263–273.

Agarwal, A.; Dekel, O.; and Xiao, L. 2010. Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback. In *Conference on Learning Theory (COLT)*, 28–40.

Agarwal, N.; and Singh, K. 2017. The price of differential privacy for online learning. In *International Conference on Machine Learning (ICML)*, 32–40.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *IEEE Foundations of Computer Science (FOCS)*, 322–331.

Awerbuch, B.; and Kleinberg, R. D. 2004. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *ACM Symposium on Theory of Computing (STOC)*, 45–53.

Bassily, R.; Smith, A.; and Thakurta, A. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE Foundations of Computer Science (FOCS)*, 464–473.

Ben-Tal, A.; and Nemirovski, A. 2001. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, volume 2. SIAM.

Bubeck, S.; Lee, Y. T.; and Eldan, R. 2017. Kernel-based methods for bandit convex optimization. In *ACM Symposium on Theory of computing (STOC)*, 72–85.

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

Bun, M.; and Steinke, T. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference (TCC)*, 635–658.

Chen, L.; Zhang, M.; and Karbasi, A. 2018. Projection-free bandit convex optimization. *arXiv preprint arXiv:1805.07474* .

Dani, V.; and Hayes, T. P. 2006. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 937–943.

Dani, V.; Kakade, S. M.; and Hayes, T. P. 2008. The price of bandit information for online optimization. In *Neural Information Processing Systems (NeurIPS)*, 345–352.

Dudik, M.; Harchaoui, Z.; and Malick, J. 2012. Lifted coordinate descent for learning with trace-norm regularization. In *Artificial Intelligence and Statistics (AISTATS)*, 327–336.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 265–284.

Dwork, C.; Naor, M.; Pitassi, T.; and Rothblum, G. N. 2010. Differential privacy under continual observation. In *ACM Symposium on Theory of Computing (STOC)*, 715–724.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9(3–4): 211–407.

Dwork, C.; and Rothblum, G. N. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887* .

Ene, A.; Nguyen, H. L.; and Vladu, A. 2021. Projection-Free Bandit Optimization with Privacy Guarantees. *arXiv preprint arXiv:2012.12138* .

Flaxman, A. D.; Kalai, A. T.; and McMahan, H. B. 2004. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint arXiv:cs/0408007* .

Garber, D.; and Hazan, E. 2013a. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666* .

Garber, D.; and Hazan, E. 2013b. Playing non-linear games with linear oracles. In *IEEE Foundations of Computer Science (FOCS)*, 420–428.

Garber, D.; and Hazan, E. 2015. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning (ICML)*, 541–549.

Garber, D.; and Kretzu, B. 2020. Improved Regret Bounds for Projection-free Bandit Convex Optimization. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2196–2206.

Hazan, E.; and Kale, S. 2012. Projection-free online learning. In *International Conference on Machine Learning (ICML)*, 521–528.

Iyengar, R.; Near, J. P.; Song, D.; Thakkar, O.; Thakurta, A.; and Wang, L. 2019. Towards practical differentially private convex optimization. In *IEEE Symposium on Security and Privacy (SP)*, 299–316.

Jaggi, M. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning (ICML)*, 427–435.

Jain, P.; Kothari, P.; and Thakurta, A. 2012. Differentially private online learning. In *Conference on Learning Theory (COLT)*, 24–1.

Lobel, I.; Leme, R. P.; and Vladu, A. 2018. Multidimensional binary search for contextual decision-making. *Operations Research* 66(5): 1346–1361.

Mao, J.; Leme, R.; and Schneider, J. 2018. Contextual pricing for lipschitz buyers. In *Neural Information Processing Systems (NeurIPS)*, 5643–5651.

McMahan, H. B.; Holt, G.; Sculley, D.; Young, M.; Ebner, D.; Grady, J.; Nie, L.; Phillips, T.; Davydov, E.; Golovin, D.; et al. 2013. Ad click prediction: a view from the trenches. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1222–1230.

Mironov, I. 2017. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, 263–275.

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical programming* 103(1): 127–152.

Shalev-Shwartz, S.; Gonen, A.; and Shamir, O. 2011. Large-scale convex minimization with a low-rank constraint. *arXiv preprint arXiv:1106.1622* .

Thakurta, A. G.; and Smith, A. 2013. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Neural Information Processing Systems (NeurIPS)*, 2733–2741.

Tropp, J. A.; et al. 2015. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning* 8(1-2): 1–230.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In *Neural Information Processing Systems (NeurIPS)*, 2722–2731.