Projection-Free Bandit Convex Optimization

Lin Chen[*1,2] **Mingrui Zhang**[*3] **Amin Karbasi**[1,2]
[1]Yale Institute for Network Science, [2]Department of Electrical Engineering,
[3]Department of Statistics and Data Science, Yale University

Abstract

In this paper, we propose the first computationally efficient projection-free algorithm for bandit convex optimization (BCO) with a general convex constraint. We show that our algorithm achieves a sublinear regret of $O(nT^{4/5})$ (where T is the horizon and n is the dimension) for any bounded convex functions with uniformly bounded gradients. We also evaluate the performance of our algorithm against baselines on both synthetic and real data sets for quadratic programming, portfolio selection and matrix completion problems.

1 INTRODUCTION

The online learning setting models a dynamic optimization process in which data becomes available in a sequential manner and the learning algorithm has to adjust and update its predictor as more data is disclosed. It can be best formulated as a repeated two-player game between a learner and an adversary as follows. At each iteration t, the learner commits to a decision $\mathbf{x}_t$ from a constraint set $\mathcal{K} \subseteq \mathbb{R}^n$. Then, the adversary selects a cost function f_t and the learner suffers the loss $f_t(\mathbf{x}_t)$ in addition to receiving feedback. In the online learning model, it is generally assumed that the learner has access to a gradient oracle for all loss functions f_t, and thus knows the loss had she chosen a different point at iteration t. The performance of an online learning algorithm is measured by a game theoretic metric known as regret which is defined as the gap between the total loss that the learner has incurred after T iterations and that of the best fixed decision in hindsight.

In online learning, we are usually interested in sublinear regret as a function of the horizon T. To this end, other structural assumptions are made. For instance, when all the loss functions f_t, as well as the constraint set $\mathcal{K}$, are convex, the problem is known as Online Convex Optimization (OCO) (Zinkevich, 2003). This framework has received a lot of attention due to its capability to model diverse problems in machine learning and statistics such as spam filtering, ad selection for search engines, and recommender systems, to name a few. It is known that the online projected gradient descent algorithm achieves a tight $O(\sqrt{T})$ regret bound (Zinkevich, 2003). However, in many modern machine learning scenarios, one of the main computational bottlenecks is the projection onto the constraint set $\mathcal{K}$. For example, in recommender systems and matrix completion, projections amount to expensive linear algebraic operations. Similarly, projections onto matroid polytopes with exponentially many linear inequalities are daunting tasks in general. This difficulty has motivated the use of projection-free algorithms (Hazan and Kale, 2012; Hazan, 2016; Chen et al., 2018) for which the most efficient one achieves $O(T^{3/4})$ regret.

In this paper, we consider a more difficult, and very often more realistic, OCO setting where the feedback is incomplete. More precisely, we consider a bandit feedback model where the only information observed by the learner at iteration t is the loss $f_t(\mathbf{x}_t)$ at the point $\mathbf{x}_t$ that she has chosen. In particular, the learner does not know the loss had she chosen a different point $\mathbf{x}_t$. Therefore, the learner has to balance between exploiting the information that she has gathered and exploring the new data. This exploration-exploitation balance has been done beautifully by (Flaxman et al., 2005) to achieve $O(T^{3/4})$ regret. With extra assumption on the loss functions (e.g., strong convexity), the regret bound has been recently improved to $\tilde{O}(T^{1/2})$ (Hazan and Li, 2016; Bubeck et al., 2015, 2017). Again, all these works either rely on the computationally expensive projection operations or inverting the Hessian matrix

[*]These authors contributed equally to this work.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

of a self-concordant barrier. In addition, regret bounds usually have a very high polynomial dependency on the dimension.

Garber and Hazan (2013) proposed a projection-free BCO algorithm that works for a *polyhedral constraint only* and attains a regret rate of $O(T^{3/4})$. A polyhedral constraint is essential: The constraint set, as an input, must be explicitly represented as the convex hull of its vertices. The construction of a local linear optimization oracle that the algorithm is based on requires computation of representing a point as a convex combination of vertices at each round. In general, however, for polyhedra defined by a system of linear inequalities, the number of vertices can increase exponentially in the dimension n, as shown by McMullen (1970). In this case, representing a point as a convex combination of vertices is particularly intractable, which significantly limits the feasibility of the algorithm in many practical problems.

In this paper, we develop the first computationally efficient projection-free BCO algorithm on a general convex constraint set, with a sublinear regret bound of $O(T^{4/5})$ on the expected regret. We also show that the dependency on the dimension is linear. The regret bounds in different OCO settings are summarized in Table 1.

Table 1: Regret bounds in various settings of adversarial online convex optimization.

	Online	**Bandit**
Projection	$O(T^{1/2})^\dagger$	$O(T^{3/4})^\ddagger$; $\tilde{O}(T^{1/2})^\sharp$
Projection-free	$O(T^{3/4})^\flat$	$O(T^{4/5})$ (**this work**)

† Zinkevich (2003)
‡ Flaxman et al. (2005)
$^\sharp$ Hazan and Li (2016); Bubeck et al. (2015, 2017)
$^\flat$ Hazan and Kale (2012)(Hazan, 2016, Alg. 24)

Our Contributions

Sublinear regret with computational efficiency. While there is a line of recent work that attains the minimax bound (Hazan and Li, 2016; Bubeck et al., 2015, 2017), these algorithms have computationally expensive parts, such as inverting the Hessian of the self-concordant barrier. In contrast to these works that seek the lowest regret bound, we try to find a computationally efficient solution that attains a sublinear regret bound. Therefore, we have to avoid computationally expensive techniques like projection, Dikin ellipsoid and self-concordant barrier. As is shown in the experiments, our algorithm is simple and effective as it only requires solving a linear optimization problem, while preserving a sublinear regret bound.

Techniques. The Frank-Wolfe (FW) algorithm may perform arbitrarily poorly with stochastic gradients even in the offline setting (Hassani et al., 2017). Since the one-point estimator of gradient has a large variance, a simple combination of online FW (Hazan and Kale, 2012) and one-point estimator (Flaxman et al., 2005) may not work. This is in fact shown empirically in Fig 1a when the loss functions are quadratic. In addition, the online FW algorithm of Hazan and Kale (2012) is infeasible in the bandit setting. Basically, in each iteration of the online FW, the linear objective is the average gradient of all previous functions at a new point x_t. Note that in the bandit setting, it is impossible to evaluate the gradient of f_i at x_t $(i < t)$, even with one-point estimators of Flaxman et al. (2005).

Our work has two major differences with (Hazan and Kale, 2012). First, to make it a bandit algorithm, our linear objective is the sum of previously estimated gradients ($\sum_{\tau=1}^{t-1} \mathbf{g}_\tau$, where $\mathbf{g}_\tau$ is the one-point estimator of $\nabla f_\tau(\mathbf{x}_\tau)$), rather than $\sum_{\tau=1}^{t-1} \nabla f_\tau(\mathbf{x}_{t-1})$. Second, we add a regularizer to stabilize the prediction.

2 PRELIMINARIES

2.1 Notation

We let $S^n \triangleq \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$ and $B^n \triangleq \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \le 1\}$ denote the unit sphere and the unit ball in the n-dimensional Euclidean space, respectively. Let $\mathbf{v}$ be a random vector. We write $\mathbf{v} \sim S^n$ and $\mathbf{v} \sim B^n$ to indicate that $\mathbf{v}$ is uniformly distributed over S^n and B^n, respectively.

For any point set $\mathcal{D} \subseteq \mathbb{R}^n$ and $\alpha > 0$, we denote $\{\mathbf{x} \in \mathbb{R}^n : \frac{1}{\alpha}\mathbf{x} \in \mathcal{D}\}$ by $\alpha\mathcal{D}$. Let $f : \mathcal{D} \to \mathbb{R}$ be a real-valued function on domain $\mathcal{D} \subseteq \mathbb{R}^n$. Its sup norm is given by $\|f\|_\infty \triangleq \sup_{\mathbf{x}\in\mathcal{D}} |f(\mathbf{x})|$. We say that the function $f : \mathcal{D} \to \mathbb{R}$ is *α-strongly convex* (Nesterov, 2003, pp. 63–64) if f is continuously differentiable, $\mathcal{D}$ is a convex set, and the following inequality holds for $\forall \mathbf{x}, \mathbf{y} \in \mathcal{D}$: $f(\mathbf{y}) \ge (\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{1}{2}\alpha\|\mathbf{y} - \mathbf{x}\|^2$. An equivalent definition of strong convexity is $(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))^\top (\mathbf{x} - \mathbf{y}) \ge \alpha\|\mathbf{x} - \mathbf{y}\|^2$, for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$. We say that f is *G-Lipschitz* if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{D}$, $\|f(\mathbf{x}) - f(\mathbf{y})\| \le G\|\mathbf{x} - \mathbf{y}\|$. In this paper, we assume that the loss functions are all convex and bounded, meaning that there is a finite M such that $\|f\|_\infty \le M$. We also assume that they are differentiable with uniformly bounded gradients, *i.e.*, there exists a finite G such that $\|\nabla f\|_\infty \le G$.

2.2 Bandit Convex Optimization

Online convex optimization is performed in a sequence of consecutive rounds, where at round t, a learner has to choose an action $\mathbf{x}_t$ from a convex decision set $\mathcal{K} \subseteq \mathbb{R}^n$. Then, an adversary chooses a loss function f_t from a family $\mathcal{F}$ of bounded convex functions. Once the action and the loss function are determined, the learner suffers a loss $f_t(\mathbf{x}_t)$. The aim is to minimize regret which is the gap between the accumulated loss and the minimum loss in hindsight. More formally, the regret of a learning algorithm $\mathcal{A}$ after T rounds is given by

$$\mathcal{R}_{\mathcal{A},T} \triangleq \sup_{\{f_1,\ldots,f_T\}\subseteq\mathcal{F}} \left\{ \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{D}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right\}.$$

In the full information setting, the learner receives the loss function f_t as a feedback (usually by having access to the gradient of f_t at any feasible decision domain). In the bandit setting, however, the feedback is limited to the loss at the point that she has chosen, *i.e.*, $f_t(\mathbf{x}_t)$. In this paper, we consider the bandit setting where the family $\mathcal{F}$ consists of bounded convex functions with uniformly bounded gradients. Under these conditions, we propose a projection-free algorithm $\mathcal{A}$ that achieves an expected regret of $\mathbb{E}[\mathcal{R}_{\mathcal{A},T}] = O(T^{4/5})$.

2.3 Smoothing

A key ingredient of our solution relies on constructing the smoothed version of loss functions. Formally, for a function f, its δ-smoothed version is defined by

$$\hat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{v}\sim B^n}[f(\mathbf{x}+\delta\mathbf{v})],$$

where $\mathbf{v}$ is drawn uniformly at random from the n-dimensional unit ball B^n. Here, δ controls the radius of the ball that the function f is averaged over. Since $\hat{f}_\delta$ is a smoothed version of f, it inherits analytical properties from f. Lemma 1 formalizes this idea.

Lemma 1 (Lemma 2.6 in (Hazan, 2016))**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be a convex, G-Lipschitz continuous function and let $\mathcal{D}_0 \subseteq \mathcal{D}$ be such that $\forall \mathbf{x} \in \mathcal{D}_0, \mathbf{v} \in S^n$, $\mathbf{x}+\delta\mathbf{v} \in \mathcal{D}$. Let $\hat{f}_\delta$ be the δ-smoothed function defined above. Then $\hat{f}_\delta$ is also convex, and $\|\hat{f}_\delta - f\|_\infty \leq \delta G$ on $\mathcal{D}_0$.*

Since $\hat{f}_\delta$ is an approximation of f, if one finds a minimizer of $\hat{f}_\delta$, Lemma 1 implies that it also minimizes f approximately. Another advantage of considering the smoothed version is that it admits one-point gradient estimates of $\hat{f}_\delta$ based on samples of f. This idea was first introduced in (Flaxman et al., 2005) for developing an online gradient descent algorithm without having access to gradients.

Lemma 2 (Lemma 6.4 in (Hazan, 2016))**.** *Let $\delta > 0$ be any fixed positive real number and $\hat{f}_\delta$ be the δ-smoothed version of function f. The following equation holds*

$$\nabla \hat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim S^n}\left[\frac{n}{\delta} f(\mathbf{x}+\delta\mathbf{u})\mathbf{u}\right]. \tag{1}$$

Lemma 2 suggests that in order to sample the gradient of $\hat{f}_\delta$ at a point $\mathbf{x}$, it suffices to evaluate f at a random point $\mathbf{x}+\delta\mathbf{u}$ around the point $\mathbf{x}$.

3 MAIN RESULTS

The first key idea of our proposed algorithm is to construct a follow-the-regularized-leader objective

$$F_t(\mathbf{x}) = \eta \sum_{\tau=1}^{t-1} \nabla f_\tau(\mathbf{x}_\tau)^\top \mathbf{x} + \|\mathbf{x}-\mathbf{x}_1\|^2. \tag{2}$$

Instead of minimizing F_t directly (as it is done in follow-the-regularized-leader algorithm), the learner first solves a linear program over the decision set $\mathcal{K}$

$$\mathbf{v}_t = \min_{\mathbf{x}\in\mathcal{K}} \{\nabla F_t(\mathbf{x}_t)\cdot\mathbf{x}\}, \tag{3}$$

and then updates its decision as follows

$$\mathbf{x}_{t+1} \leftarrow (1-\sigma_t)\mathbf{x}_t + \sigma_t \mathbf{v}_t. \tag{4}$$

Note that minimizing F_t requires solving a quadratic optimization problem, which is as computationally prohibitive as a projection operation. In contrast, since the update in (4) is a convex combination between $\mathbf{v}_t$ and $\mathbf{x}_t$, the iterates always lie inside the convex decision set $\mathcal{K}$, thus no projection is needed. This is the main idea behind the online conditional gradient algorithm (Algorithm 24 in (Hazan, 2016)). In the bandit setting (the focus of this paper), the gradients $\nabla f_\tau(\mathbf{x}_\tau)$ are unavailable, hence the learner cannot perform steps (2) and (3). To tackle this issue, we introduce the second ingredient of our algorithm, namely, the smoothing and one-point gradient estimates (Flaxman et al., 2005). Formally, at the t-th iteration, rather than selecting $\mathbf{x}_t$, the learner plays a random point $\mathbf{y}_t$ that is δ-close to $\mathbf{x}_t$ and in return observes the cost $f_t(\mathbf{y}_t)$. As shown in Lemma 2, $f_t(\mathbf{y}_t)$ can be used to construct an unbiased estimate $\mathbf{g}_t$ for the gradient of the δ-smoothed version of f_t at point $\mathbf{x}_t$, *i.e.*, $\mathbb{E}[\mathbf{g}_t] = \nabla \hat{f}_{t,\delta}(\mathbf{x}_t)$, where $\hat{f}_{t,\delta}(\mathbf{x}_t) \triangleq \mathbb{E}_{\mathbf{v}\sim B^n}[f_t(\mathbf{x}_t+\delta\mathbf{v})]$. This observation suggests that we can replace $\nabla f_t(\mathbf{x}_t)$ by $\mathbf{g}_t$ in the follow-the-regularized-leader objective (2) to obtain a variant that relies on the one-point gradient estimate, *i.e.*,

$$F_t(\mathbf{x}) = \eta \sum_{\tau=1}^{t-1} \mathbf{g}_\tau^\top \mathbf{x} + \|\mathbf{x}-\mathbf{x}_1\|^2. \tag{5}$$

Note that forming $F_t(\mathbf{x})$ in (5) is fully realizable for a learner in a bandit setting. The full description of our algorithm is outlined in Algorithm 1. Even though the objective function $F_t(\mathbf{x})$ relies on the unbiased estimates of the smoothed versions of f_t (rather than f_t itself), it is not far off from the original objective (shown in (2)) if the distance between the random point $\mathbf{y}_t$ and the point $\mathbf{x}_t$ is properly chosen. Therefore, minimizing the sum of smoothed versions of f_t (as it is done by Algorithm 1) will end up minimizing the actual regret. This intuition is formally proven in Theorem 1. Without loss of generality, we assume additionally that the constraint $\mathcal{K}$ contains a ball of radius r centered at the origin (this is always achievable by shrinking the constraint set as long as it has a non-empty interior).

Algorithm 1 Projection-Free Bandit Convex Optimization

Input: horizon T, constraint set $\mathcal{K}$
Output: $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$

1: $\mathbf{x}_1 \in (1-\alpha)\mathcal{K}$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad \mathbf{y}_t \leftarrow \mathbf{x}_t + \delta \mathbf{u}_t$, where $\mathbf{u}_t \sim S^n$
4: $\quad$ Play $\mathbf{y}_t$ and observe $f_t(\mathbf{y}_t)$
5: $\quad \mathbf{g}_t \leftarrow \frac{n}{\delta} f_t(\mathbf{y}_t)\mathbf{u}_t$ ▷ $\mathbf{g}_t$ is an unbiased estimator of $\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)$
6: $\quad F_t(\mathbf{x}) \leftarrow \eta \sum_{\tau=1}^{t-1} \mathbf{g}_\tau^\top \mathbf{x} + \|\mathbf{x} - \mathbf{x}_1\|^2$
7: $\quad \mathbf{v}_t \leftarrow \arg\min_{\mathbf{x} \in (1-\alpha)\mathcal{K}} \{\nabla F_t(\mathbf{x}_t) \cdot \mathbf{x}\}$ ▷ Solve a linear optimization problem
8: $\quad \mathbf{x}_{t+1} \leftarrow (1-\sigma_t)\mathbf{x}_t + \sigma_t \mathbf{v}_t$
9: **end for**

Theorem 1 (**Proof in Section 5**). *Assume that for every $t \in \mathbb{N}_{\geq 1}$, f_t is convex, $\|f_t\|_\infty \leq M$ on $\mathcal{K}$, $\sup_{\mathbf{x}\in\mathcal{K}} \|\nabla f_t(\mathbf{x})\| \leq G$, $rB^n \subseteq \mathcal{K} \subseteq RB^n$, and that the diameter of $\mathcal{K}$ is $D < \infty$. If we set $\eta = \frac{D}{\sqrt{2}nM} T^{-4/5}$, $\sigma_t = t^{-2/5}$, $\delta = cT^{-1/5}$, and $\alpha = \delta/r < 1$ in Algorithm 1, where $c > 0$ is a constant, we have $\mathbf{y}_t \in \mathcal{K}, \forall 1 \leq t \leq T$. Moreover, the expected regret $\mathbb{E}[\mathcal{R}_{\mathcal{A},T}]$ up to horizon T is at most*

$$\frac{\sqrt{2}nMD}{c^2} T^{3/5} + (\sqrt{2}nMD + \frac{5\sqrt{2}}{4} DG + 3cG + cRG/r) T^{4/5}.$$

Note that the regret bound of Algorithm 1 depends linearly on the dimension n.

A minor drawback of Algorithm 1 is that it requires the knowledge of the horizon T. This problem can be easily circumvented via the doubling trick while preserving the regret bound of Theorem 1. The doubling trick was first proposed in (Auer et al., 1995) and its key idea is to invoke the base algorithm repeatedly with a doubling horizon. Algorithm 2 outlines an *anytime* algorithm for BCO using the doubling trick. Theorem 2 shows that for any $t \geq 1$, the expected regret of Algorithm 2 by the end of the t-th iteration is bounded by $O(t^{4/5})$.

Algorithm 2 Anytime Projection-Free Bandit Convex Optimization

Input: constraint set $\mathcal{K}$
Output: $\mathbf{y}_1, \mathbf{y}_2, \ldots$

1: **for** $m = 0, 1, 2, \ldots$ **do**
2: $\quad$ Run Algorithm 1 with horizon 2^m from the 2^m-th iteration (inclusive) to the $(2^{m+1}-1)$-th iteration (inclusive).
3: $\quad$ Let $\mathbf{y}_{2^m}, \ldots, \mathbf{y}_{2^{m+1}-1}$ be the points that Algorithm 1 selects for the objectives $f_{2^m}, \ldots, f_{2^{m+1}-1}$.
4: **end for**

Theorem 2 (**Proof in Appendix D**). *If the regret bound of Algorithm 1 for horizon T is $\beta T^{4/5}$, then for any $t \geq 1$, the expected regret of Algorithm 2 by the end of the t-th iteration is at most*

$$\mathbb{E}[\mathcal{R}_{\mathcal{A},T}] = \frac{\beta}{1 - 2^{-4/5}} (t+1)^{4/5} = O(t^{4/5}).$$

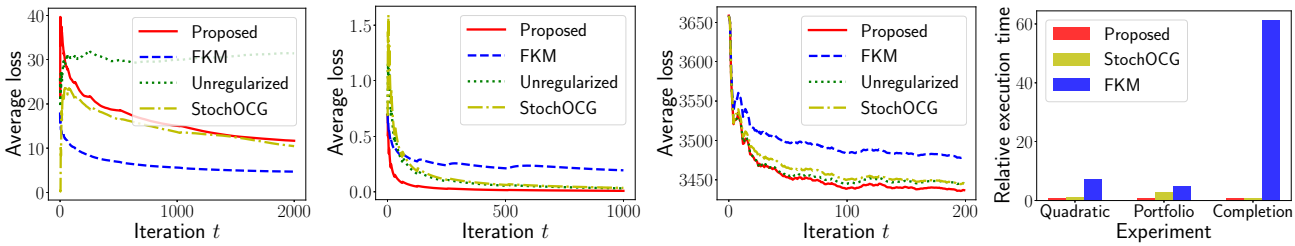
4 EXPERIMENTS

In our set of experiments, we compare Algorithm 2 with the following baselines: (1) FKM: Online projected gradient descent with spherical gradient estimators (Flaxman et al., 2005). (2) Unregularized: A variant of our proposed algorithm without the regularizer $\|\mathbf{x} - \mathbf{x}_1\|^2$ in line 6 of Algorithm 1. (3) StochOCG: Online conditional gradient (Hazan, 2016) with stochastic gradients (not a bandit algorithm). Such stochastic gradients are formed by adding Gaussian noise with standard deviation n to the exact gradients.

The anytime version of the algorithms (obtained via the doubling trick) is used. Therefore the horizon T is unknown to the algorithms. Since the standard deviation of the point estimate used in FKM and our proposed method is proportional to the dimension n, the standard deviation of the Gaussian noise in StochOCG is set to n to make the noise in the gradients comparable. Note that Hazan (2016) assumed access to exact gradients. It remains unknown whether it is robust to a noisy gradients.

We performed three sets of experiments in total. In all of them we report the average loss defined as $\mathbb{E}[\sum_{t=1}^{T} f_t(\mathbf{x}_t)]/T$.

Quadratic Programming: In the first experiment, the loss functions are quadratic, *i.e.*, $f_t(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{G}_t^\top \mathbf{G}_t \mathbf{x} + \mathbf{w}_t^\top \mathbf{x}$. Each entry of $\mathbf{G}_t$ and $\mathbf{w}_t$ is sampled from the standard normal distribution. The convex constraint of this problem is a polytope $\{\mathbf{x} : \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \mathbf{A}\mathbf{x} \leq \mathbf{1}\}$ and each entry of $\mathbf{A}$ is sampled from the uniform distribution on $[0, 1]$. The average loss is



(a) Quadratic programming (b) Portfolio selection (c) Matrix completion (d) Relative execution time

Figure 1: In Figs. 1a to 1c, we show the average loss versus the number of iterations in the three sets of experiments. The relative execution time is shown in Fig. 1d, where the execution time of the proposed algorithm is set to 1.

illustrated in Fig. 1a. We observe that the average loss of our proposed algorithm declines as the number of iterations increases. This agrees with the theoretical sublinear regret bound. StochOCG has a similar performance while FKM exhibits the lowest loss. In contrast, the loss of Unregularized appears to be linear which shows the significance of regularization to achieve low regret. This observation also suggests that simply combining (Hazan and Kale, 2012) and smoothing may not work in practice.

Portfolio Selection: For this experiment, we randomly select $n = 100$ stocks from Standard & Poor's 500 index component stocks and consider their prices during the business days between February 18th, 2013 and November 27th, 2017. We follow the formulation in (Hazan, 2016, Section 1.2). Let $\mathbf{r}_t \in \mathbb{R}^n$ be a vector such that $\mathbf{r}_t(i)$ is the ratio of the price of stock i on day $t+1$ to its price on day t. An investor is trying maximize her wealth by investing on different stock options. If W_t denotes her wealth on day t, then we have the following recursion: $W_{t+1} = W_t \cdot \mathbf{r}_t^\top \mathbf{x}_t$. After T days of investments, the total wealth will be $W_T = W_1 \cdot \prod_{t=1}^T \mathbf{r}_t^\top \mathbf{x}_t$. To maximize the wealth, the investor has to maximize $\sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}_t)$, or equivalently minimize its negation. Thus, we can define $f_t(\mathbf{x}_t) \triangleq -\log(\mathbf{r}_t^\top \mathbf{x}_t)$. FKM requires that the constraint set contains the unit ball. To this end, we set $\mathbf{y}_t = 2n\mathbf{x}_t - 1$ so that $\mathbf{y}_t$ lies in an enlarged region $\Delta_n' \triangleq \{\mathbf{y} \in \mathbb{R}^n : -1 \le \mathbf{y}(i) \le 2n-1, \sum_{i=1}^n \mathbf{y}(i) \le n\}$. In addition, the objective functions f_t are viewed as functions of $\mathbf{y}_t$ rather than $\mathbf{x}_t$. The average losses versus the number of iterations are presented in Fig. 1b. Our proposed algorithm has the lowest loss in this set of experiments while FKM has the largest.

Matrix Completion: Let $\{\mathbf{M}_t\}_{t=1}^T$ be symmetric positive semi-definite (PSD) matrices, where $\mathbf{M}_t = \mathbf{N}_t^\top \mathbf{N}_t$ and every entry of $\mathbf{N}_t \in \mathbb{R}^{k\times n}$ obeys the standard normal distribution. At each iteration, half of the entries of $\mathbf{M}_t$ are observed. We set $n = 20$ and $k = 18$. We denote the entries of $\mathbf{M}_t$ disclosed at the t-th iteration by O_t. We want to minimize $f_t(\mathbf{X}_t) \triangleq \frac{1}{2}\sum_{(i,j)\in O_t}(\mathbf{X}_t[i,j] - \mathbf{M}_t[i,j])^2$ subject to $\|\mathbf{X}_t\|_* \le k$, where $\mathbf{X}_t$ is of the same shape as $\mathbf{M}_t$ and $\|\cdot\|_*$ denotes the nuclear norm. The nuclear norm constraint is a standard convex relaxation of the rank constraint $\mathrm{rank}(\mathbf{X}) \le k$. The linear optimization step in Line 7 of Algorithm 1 has a closed-form solution $\mathbf{v}_t = k\mathbf{v}_{\max}\mathbf{v}_{\max}^\top$, where $\mathbf{v}_{\max}$ is the eigenvector of the largest eigenvalue of $-\nabla f_t(\mathbf{X}_t)$ (Hazan, 2016, Section 7.3.1). The largest eigenvector can be computed very efficiently using power iterations, whilst it is extremely costly to perform projection onto a convex subset of the space of PSD matrices. As shown in Fig. 1d, the efficiency of the proposed algorithm is 61 times that of the projection-based FKM algorithm. The average loss of the algorithms is shown in Fig. 1c. Our proposed algorithm outperforms the other baselines while FKM suffers the largest loss.

We also observe rises of the curves at their initial stage in Fig. 1. They are due to the doubling trick (Algorithm 2) and a small denominator of the average loss. The unknown horizon is divided into epochs with a doubling size (1, 2, 4, and so forth). When the algorithm starts a new epoch, everything is reset and the algorithm learns from scratch. Furthermore, the denominator of the average loss is small (it is initially 1, and then becomes $2, 3, 4, \dots$) at the initial stage. Therefore, due to frequent resets and a small denominator, the behavior is less stable. As the epoch size and denominator grow, the average loss declines steadily.

While our proposed algorithm outperforms FKM in Figs. 1b and 1c, this does not contradict the orders of regret presented in Table 1. According to (Flaxman et al., 2005), the $O(T^{3/4})$ bound holds when $T \ge (3Rn/(2r))^2$. In portfolio selection, since $R/r = \sqrt{4n^2 - 3n}$ and $n = 100$, we have $(3Rn/(2r))^2 > 8.9 \times 10^8$, which is much larger than our horizon ($= 1000$). In matrix completion, since the dimension $n = (20\times 20 - 20)/2 = 190$, we have $(3Rn/(2r))^2 \ge (3n/2)^2 > 8.1 \times 10^4$, greatly exceeding our horizon ($= 200$).

The execution time is shown in Fig. 1d. It was mea-

sured on eight Intel Xeon E5-2660 V2 cores and the algorithms were implemented in Julia. 50 repeated experiments were run in parallel. It can be observed that our proposed algorithm runs significantly faster than the FKM algorithm (mostly by avoiding the projection steps). Specifically, its efficiency is almost 7 times, 5 times, and 61 times that of the FKM algorithm in the three sets of experiments, respectively. StochOCG requires computation of gradients and is also slower than the proposed algorithm.

5 PROOF OF THEOREM 1

First we show $\mathbf{y}_t \in \mathcal{K}$. Since $\mathbf{v}_t \in (1-\alpha)\mathcal{K}$, $\mathbf{x}_1 \in (1-\alpha)\mathcal{K}$ and $\mathbf{x}_{t+1} = (1-\sigma_t)\mathbf{x}_t + \sigma_t \mathbf{v}_t$, by induction and the convexity of $\mathcal{K}$, we have $\mathbf{x}_t \in (1-\alpha)\mathcal{K}$ for every t. Recall that $\mathbf{y}_t = \mathbf{x}_t + \delta \mathbf{u}_t$, where $\mathbf{u}_t \in S^n$ and $\alpha = \delta/r$. Since $\mathcal{K}$ is convex and $rS^n \subseteq rB^n \subseteq \mathcal{K}$, we have $\mathbf{y}_t \in (1-\alpha)\mathcal{K} + \alpha r S^n \subseteq (1-\alpha)\mathcal{K} + \alpha\mathcal{K} = \mathcal{K}$.

Let $\mathbf{x}_t^* \triangleq \arg\min_{\mathbf{x}\in(1-\alpha)\mathcal{K}} F_t(\mathbf{x})$ and $\hat{f}_{t,\delta}(\mathbf{x}_t) \triangleq \mathbb{E}_{\mathbf{v}\sim B^n}[f_t(\mathbf{x}_t + \delta\mathbf{v})]$. The first step is to derive a bound on $\sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z})$. We need the following lemma.

Lemma 3 (Lemma 2.3 in (Shalev-Shwartz, 2012))**.** *Let $\mathbf{w}_1, \mathbf{w}_2, \ldots$ be a sequence of vectors in $(1-\alpha)\mathcal{K}$ such that $\forall t, \mathbf{w}_t = \arg\min_{\mathbf{w}\in(1-\alpha)\mathcal{K}} \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$. Then for every $\mathbf{z} \in (1-\alpha)\mathcal{K}$, we have $\sum_{t=1}^T (f_t(\mathbf{w}_t) - f_t(\mathbf{z})) \le R(\mathbf{z}) - R(\mathbf{w}_1) + \sum_{t=1}^T (f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}))$.*

By Lemma 3 and in light of the fact that $\mathbf{x}_1^* = \mathbf{x}_1$, $\forall \mathbf{z} \in (1-\alpha)\mathcal{K}$, we have

$$\begin{aligned}
&\sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z}) \\
\le& \|\mathbf{z} - \mathbf{x}_1\|^2/\eta - \|\mathbf{x}_1^* - \mathbf{x}_1\|^2/\eta + \sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{x}_{t+1}^*) \\
=& \|\mathbf{z} - \mathbf{x}_1\|^2/\eta + \sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{x}_{t+1}^*).
\end{aligned} \tag{6}$$

Let $\mathcal{F}_t$ be the σ-field generated by $\mathbf{x}_1, \mathbf{g}_1, \mathbf{x}_2, \mathbf{g}_2, \ldots, \mathbf{x}_{t-1}, \mathbf{g}_{t-1}, \mathbf{x}_t$. Note that $\mathbf{x}_t^*$ is a function of $\mathbf{g}_1, \ldots, \mathbf{g}_{t-1}$ and thus measurable with respect to $\mathcal{F}_t$. Therefore we have $\mathbb{E}[\mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z})] = \mathbb{E}[\mathbb{E}[\mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z})|\mathcal{F}_t]] = \mathbb{E}[\mathbb{E}[\mathbf{g}_t|\mathcal{F}_t]^\top(\mathbf{x}_t^* - \mathbf{z})] = \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t^* - \mathbf{z})]$. To bound the second term on the right-hand side of (6), note that $\mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{x}_{t+1}^*) \le 2\eta\|\mathbf{g}_t\|^2$ (we will show it in Appendix A). Therefore we have $\sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{x}_{t+1}^*) \le 2\eta \sum_{t=1}^T \|\mathbf{g}_t\|^2 \le 2\eta n^2 M^2 T/\delta^2$. Combining it with (6), we deduce $\sum_{t=1}^T \mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z}) \le D^2/\eta + 2\eta n^2 M^2 T/\delta^2$.

Since

$$\begin{aligned}
\sum_{t=1}^T \mathbb{E}[f_t(\mathbf{y}_t) - f_t(\mathbf{z})] =& \sum_{t=1}^T \mathbb{E}[f_t(\mathbf{y}_t) - f_t(\mathbf{x}_t)] \\
&+ \sum_{t=1}^T \mathbb{E}[f_t(\mathbf{x}_t) - f_t(\mathbf{z})],
\end{aligned} \tag{7}$$

and the norm of the gradient of f_t is assumed to be at most G

$$\sum_{t=1}^T \mathbb{E}[f_t(\mathbf{y}_t) - f_t(\mathbf{x}_t)] \le \delta T G, \tag{8}$$

we only need to obtain an upper bound of the second term on the right hand side of (7), which is

$$\begin{aligned}
&\sum_{t=1}^T \mathbb{E}[f_t(\mathbf{x}_t) - f_t(\mathbf{z})] \\
=& \mathbb{E}[\sum_{t=1}^T (\hat{f}_{t,\delta}(\mathbf{x}_t) - \hat{f}_{t,\delta}(\mathbf{z})) + \sum_{t=1}^T (f_t(\mathbf{x}_t) - \hat{f}_{t,\delta}(\mathbf{x}_t)) \\
&\quad - \sum_{t=1}^T (f_t(\mathbf{z}) - \hat{f}_{t,\delta}(\mathbf{z}))] \\
\overset{(a)}{\le}& \mathbb{E}\left[\sum_{t=1}^T (\hat{f}_{t,\delta}(\mathbf{x}_t) - \hat{f}_{t,\delta}(\mathbf{z}))\right] + 2\delta G T \\
\overset{(b)}{\le}& \sum_{t=1}^T \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{z})] + 2\delta G T.
\end{aligned}$$

Inequality (a) is due to Lemma 1. We used the convexity of $\hat{f}_{t,\delta}$ in (b). We split $\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{z})$ into $\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t^* - \mathbf{z}) + \nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}_t^*)$ and thus obtain

$$\begin{aligned}
&\sum_{t=1}^T \mathbb{E}[f_t(\mathbf{x}_t) - f_t(\mathbf{z})] \\
\le& \sum_{t=1}^T \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t^* - \mathbf{z})] \\
&\quad + \sum_{t=1}^T \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}_t^*)] + 2\delta G T \\
=& \sum_{t=1}^T \mathbb{E}[\mathbf{g}_t^\top(\mathbf{x}_t^* - \mathbf{z})] + \sum_{t=1}^T \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}_t^*)] \\
&\quad + 2\delta G T \\
\le& D^2/\eta + 2\eta n^2 M^2 T/\delta^2 + \sum_{t=1}^T \mathbb{E}[\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}_t^*)] \\
&\quad + 2\delta G T.
\end{aligned} \tag{9}$$

The next step is to bound $\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}_t^*)$. To this end, we need the auxiliary inequality below.

Lemma 4 (**Proof in Appendix B**). *The inequality* $-4t^{2/5}(t+1)^{2/5}+4t^{4/5}-2t^{1/5}(t+1)^{1/5}+3(t+1)^{2/5} \geq 0$ *holds for any* $t = 1, 2, 3, \ldots$.

In light of the inequality, we have

$$\begin{aligned} &t^{3/5}(t+1)^{1/5}\left(\frac{3}{2t^{4/5}} - \frac{2}{t^{2/5}} + \frac{2}{(t+1)^{2/5}}\right) \\ =&\frac{-4t^{2/5}(t+1)^{2/5} + 4t^{4/5} + 3(t+1)^{2/5}}{2t^{1/5}(t+1)^{1/5}} \geq 1. \end{aligned}$$

By algebraic manipulation, we see

$$\begin{aligned} &\frac{2\sigma_{t+1} - 2\sigma_t + (3/2)\sigma_t^2}{\sqrt{2\sigma_{t+1}}} \\ =&\frac{1}{\sqrt{2}}(t+1)^{1/5}\left(\frac{3}{2t^{4/5}} - \frac{2}{t^{2/5}} + \frac{2}{(t+1)^{2/5}}\right) \qquad (10) \\ \geq&\frac{1}{\sqrt{2}}t^{-3/5}. \end{aligned}$$

If $1 \leq t \leq T$, we deduce

$$\frac{1}{\sqrt{2}}t^{-3/5} \geq \frac{1}{\sqrt{2}}T^{-3/5} = \frac{\eta n M}{\delta D} \geq \frac{\eta}{D}\|\mathbf{g}_s\|, \quad \forall 1 \leq s \leq T. \tag{11}$$

Combining (10) and (11), we deduce

$$\eta \leq D\frac{2\sigma_{t+1} - 2\sigma_t + (3/2)\sigma_t^2}{\|\mathbf{g}_{t+1}\|\sqrt{2\sigma_{t+1}}}, \quad \forall 1 \leq t \leq T.$$

The above inequality is equivalent to

$$\begin{aligned} &2(1-\sigma_t)D^2\sigma_t + \frac{D^2}{2}\sigma_t^2 + (\eta\|\mathbf{g}_{t+1}\|/2)^2 \\ \leq&2D^2\sigma_{t+1} + (\eta\|\mathbf{g}_{t+1}\|/2)^2 - \eta\|\mathbf{g}_{t+1}\|\sqrt{2D^2\sigma_{t+1}}. \end{aligned}$$

Before taking the square root, we need Lemma 5.

Lemma 5 (**Proof in Appendix C**). *Under the assumptions of Theorem 1,* $\sqrt{2D^2\sigma_{t+1}} \geq \eta\|\mathbf{g}_{t+1}\|/2$ *holds for any* $1 \leq t \leq T$.

Since $\sqrt{2D^2\sigma_{t+1}} \geq \eta\|\mathbf{g}_{t+1}\|/2$, taking the square root of both sides, we obtain $\sqrt{2(1-\sigma_t)D^2\sigma_t + \frac{D^2}{2}\sigma_t^2 + (\eta\|\mathbf{g}_{t+1}\|/2)^2} \leq \sqrt{2D^2\sigma_{t+1}} - \eta\|\mathbf{g}_{t+1}\|/2$, which is equivalent to

$$\begin{aligned} &\sqrt{2(1-\sigma_t)D^2\sigma_t + \frac{D^2}{2}\sigma_t^2 + (\eta\|\mathbf{g}_{t+1}\|/2)^2} + \eta\|\mathbf{g}_{t+1}\|/2 \\ \leq&\sqrt{2D^2\sigma_{t+1}}. \end{aligned} \tag{12}$$

If $h_t(\mathbf{x}) \triangleq F_t(\mathbf{x}) - F_t(\mathbf{x}_t^*)$ and $h_t \triangleq h_t(\mathbf{x}_t)$, we have

$$\begin{aligned} &h_t(\mathbf{x}_{t+1}) = F_t(\mathbf{x}_{t+1}) - F_t(\mathbf{x}_t^*) \\ =&F_t((1-\sigma_t)\mathbf{x}_t + \sigma_t\mathbf{v}_t) - F_t(\mathbf{x}_t^*) \\ =&F_t(\mathbf{x}_t + \sigma_t(\mathbf{v}_t - \mathbf{x}_t)) - F_t(\mathbf{x}_t^*) \\ \leq&F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) + \sigma_t\nabla F_t(\mathbf{x}_t)^\top(\mathbf{v}_t - \mathbf{x}_t) + D^2\sigma_t^2/2 \\ \leq&F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) + \sigma_t\nabla F_t(\mathbf{x}_t)^\top(\mathbf{x}_t^* - \mathbf{x}_t) + D^2\sigma_t^2/2 \\ \leq&F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*) + \sigma_t(F_t(\mathbf{x}_t^*) - F_t(\mathbf{x}_t)) + D^2\sigma_t^2/2 \\ =&(1-\sigma_t)(F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*)) + D^2\sigma_t^2/2 \\ =&(1-\sigma_t)h_t + D^2\sigma_t^2/2. \end{aligned}$$

By the definition of h_t and F_t and in light of the fact that $\mathbf{x}_t^*$ is the minimizer of F_t, we obtain

$$\begin{aligned} h_{t+1} =&F_t(\mathbf{x}_{t+1}) - F_t(\mathbf{x}_{t+1}^*) + \eta\mathbf{g}_{t+1}(\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^*) \\ \leq&F_t(\mathbf{x}_{t+1}) - F_t(\mathbf{x}_t^*) + \eta\mathbf{g}_{t+1}(\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^*) \\ =&h_t(\mathbf{x}_{t+1}) + \eta\mathbf{g}_{t+1}(\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^*) \\ \leq&h_t(\mathbf{x}_{t+1}) + \eta\|\mathbf{g}_{t+1}\|\|\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^*\|. \end{aligned}$$

Notice that F_t is 2-strongly convex and that $\mathbf{x}_t^*$ is the minimizer of F_t. We have $\|\mathbf{x} - \mathbf{x}_t^*\|^2 \leq F_t(\mathbf{x}) - F_t(\mathbf{x}_t^*)$. Therefore we obtain

$$\begin{aligned} h_{t+1} \leq&(1-\sigma_t)h_t + D^2\sigma_t^2/2 \\ &+ \eta\|\mathbf{g}_{t+1}\|\sqrt{F_{t+1}(\mathbf{x}_{t+1}) - F_{t+1}(\mathbf{x}_{t+1}^*)} \\ =&(1-\sigma_t)h_t + D^2\sigma_t^2/2 + \eta\|\mathbf{g}_{t+1}\|\sqrt{h_{t+1}}. \end{aligned}$$

We will show $h_\tau \leq 2D^2\sigma_\tau$ holds for $\forall 1 \leq \tau \leq T$ by induction. Since $h_1 = F_1(\mathbf{x}_1) - F_1(\mathbf{x}_1^*) = 0$, it holds if $t = 1$. Assume that it holds for $\tau = t$. Now we set $\tau = t+1$. By the induction hypothesis, we have

$$h_{t+1} \leq 2(1-\sigma_t)D^2\sigma_t + D^2\sigma_t^2/2 + \eta\|\mathbf{g}_{t+1}\|\sqrt{h_{t+1}}.$$

By completing the square, we obtain $(\sqrt{h_{t+1}} - \eta\|\mathbf{g}_{t+1}\|/2)^2 \leq 2(1-\sigma_t)D^2\sigma_t + D^2\sigma_t^2/2 + (\eta\|\mathbf{g}_{t+1}\|/2)^2$. Therefore,

$$\begin{aligned} \sqrt{h_{t+1}} \leq&\sqrt{2(1-\sigma_t)D^2\sigma_t + D^2\sigma_t^2/2 + (\eta\|\mathbf{g}_{t+1}\|/2)^2} \\ &+ \eta\|\mathbf{g}_{t+1}\|/2. \end{aligned}$$

By (12), the right-hand side is at most $\sqrt{2D^2\sigma_{t+1}}$. Thus we conclude that $h_{t+1} \leq 2D^2\sigma_{t+1}$. Then we are able to bound $\|\mathbf{x}_t - \mathbf{x}_t^*\|$ as follows: $\|\mathbf{x}_t - \mathbf{x}_t^*\| \leq \sqrt{F_t(\mathbf{x}_t) - F_t(\mathbf{x}_t^*)} \leq \sqrt{2D^2\sigma_t} = \sqrt{2}Dt^{-1/5}$. By (9), and since $\|\nabla \hat{f}_{t,\delta}(\mathbf{x}_t)\| \leq \mathbb{E}_{\mathbf{v}\sim B^n}[\|\nabla f_t(\mathbf{x}_t + \delta\mathbf{v})\|] \leq G$,

we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}[f_t(\mathbf{x}_t)-f_t(\mathbf{z})]\\
\leq& D^2/\eta+2\eta n^2M^2T/\delta^2+G\sum_{t=1}^{T}\mathbb{E}[\|\mathbf{x}_t-\mathbf{x}_t^*\|]+2\delta GT\\
\overset{(a)}{\leq}&\sqrt{2}nMDT^{4/5}+\frac{\sqrt{2}nMD}{c^2}T^{3/5}+\frac{5\sqrt{2}}{4}DGT^{4/5}\\
&+2cGT^{4/5}\\
=&\frac{\sqrt{2}nMD}{c^2}T^{3/5}+(\sqrt{2}nMD+\frac{5\sqrt{2}}{4}DG+2cG)T^{4/5},
\end{aligned}$$

where we use $\sum_{t=1}^{T}t^{-1/5}\leq\frac{5}{4}T^{4/5}$ in (a). Adding (8) to the above inequality, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}[f_t(\mathbf{y}_t)-f_t(\mathbf{z})] \qquad (13)\\
\leq&\frac{\sqrt{2}nMD}{c^2}T^{3/5}+(\sqrt{2}nMD+\frac{5\sqrt{2}}{4}DG+3cG)T^{4/5}.
\end{aligned}$$

Let $\mathbf{x}^*\triangleq\arg\min_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}f_t(\mathbf{x})$ and $\Pi(\mathbf{x}^*)\triangleq\arg\min_{\mathbf{x}\in(1-\alpha)\mathcal{K}}\|\mathbf{x}-\mathbf{x}^*\|$. We have $\|\mathbf{x}^*-\Pi(\mathbf{x}^*)\|\leq\|\mathbf{x}^*-(1-\alpha)\mathbf{x}^*\|\leq\alpha R$. If we set $\mathbf{z}=\Pi(\mathbf{x}^*)$ in (13), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}[f_t(\mathbf{y}_t)-f_t(\mathbf{x}^*)]\\
=&\sum_{t=1}^{T}\mathbb{E}[f_t(\mathbf{y}_t)-f_t(\Pi(\mathbf{x}^*))+f_t(\Pi(\mathbf{x}^*))-f_t(\mathbf{x}^*)]\\
\leq&\frac{\sqrt{2}nMD}{c^2}T^{3/5}+(\sqrt{2}nMD+\frac{5\sqrt{2}}{4}DG+3cG)T^{4/5}\\
&+\alpha RGT.
\end{aligned}$$

Since $\alpha=\delta/r$, we conclude that the regret is at most

$$\frac{\sqrt{2}nMD}{c^2}T^{3/5}+(\sqrt{2}nMD+\frac{5\sqrt{2}}{4}DG+3cG+cRG/r)T^{4/5}.$$

6 RELATED WORK

Zinkevich (2003) introduced the online convex optimization (OCO) problem and proposed online gradient descent. OCO generalizes existing models of online learning, including the universal portfolios model (Cover, 1991) and prediction from expert advice (Littlestone and Warmuth, 1994). For strongly convex functions, an algorithm that achieves a logarithmic regret was proposed in (Hazan et al., 2007). Regularization-based methods applied to OCO problems were investigated in (Grove et al., 2001; Kivinen and Warmuth, 1998). The follow-the-perturbed-leader algorithm was introduced and analyzed in (Kalai and Vempala, 2005). Thereafter, the follow-the-regularized-leader (FTRL) was independently considered in (Shalev-Shwartz, 2007; Shalev-Shwartz and Singer, 2007) and (Abernethy et al., 2008). Hazan and Kale (2010) showed the equivalence of FTRL and online mirror descent.

For projection-free convex optimization, the Frank-Wolfe algorithm (*i.e.*, the conditional gradient method) was originally proposed in (Frank and Wolfe, 1956), and was further analyzed in (Jaggi, 2013). The online conditional gradient method was investigated in (Hazan and Kale, 2012). A distributed version was proposed in (Zhang et al., 2017). Conditional gradient methods are very sensitive to noisy gradients. This issue was recently resolved in centralized (Mokhtari et al., 2018) and online settings (Chen et al., 2018).

A special case of bandit convex optimization (BCO) with linear objectives was studied in (Awerbuch and Kleinberg, 2008; Bubeck et al., 2012a; Karnin and Hazan, 2014). The general problem of BCO was considered in (Flaxman et al., 2005) and was further studied in (Dani et al., 2008; Agarwal et al., 2011; Bubeck et al., 2012b; Bubeck and Eldan, 2016). A near-optimal regret algorithm for the BCO problem with strongly-convex and smooth losses was introduced in (Hazan and Levy, 2014), while BCO with Lipschitz-continuous convex losses was analyzed in (Kleinberg, 2005). Regret rate $\tilde{O}(T^{2/3})$ was achieved in (Saha and Tewari, 2011) for convex and smooth loss functions, and in (Agarwal et al., 2010) for strongly-convex loss functions, and was improved to $\tilde{O}(T^{5/8})$ in (Dekel et al., 2015). For strongly-convex and smooth loss functions, a lower bound of $\Omega(\sqrt{T})$ was attained in (Shamir, 2013). Bubeck et al. (2017) proposed the first poly$(n)\sqrt{T}$-regret algorithm whose running time is polynomial in horizon T. Zeroth-order optimization is also relevant to BCO. Interested readers are referred to (Conn et al., 2009; Duchi et al., 2015; Yu et al., 2016).

7 CONCLUSION

We presented the first computationally efficient projection-free BCO algorithm that requires no knowledge of the horizon T and achieve an $O(nT^{4/5})$ regret bound. Our experimental results show that our proposed algorithm exhibits a sublinear regret and runs significantly faster than the other baselines.

Acknowledgements

LC was supported by Google PhD Fellowship. AK was supported by AFOSR YIP award (FA9550-18-1-0160).

References

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274, 2008.

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

Alekh Agarwal, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. In *NIPS*, pages 1035–1043, 2011.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *FOCS*, pages 322–331. IEEE, 1995.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *COLT*, pages 583–589, 2016.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *COLT*, volume 23, pages 41.1–41.14, 2012a.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012b.

Sébastien Bubeck, Ofer Dekel, Tomer Koren, and Yuval Peres. Bandit convex optimization:$\sqrt{T}$ regret in one dimension. In *COLT*, pages 266–278, 2015.

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *STOC*, pages 72–85. ACM, 2017.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *ICML*, page to appear, 2018.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

Thomas M Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

Varsha Dani, Sham M Kakade, and Thomas P Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2008.

Ofer Dekel, Ronen Eldan, and Tomer Koren. Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *NIPS*, pages 2926–2934, 2015.

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, pages 385–394, 2005.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

Dan Garber and Elad Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

Adam J Grove, Nick Littlestone, and Dale Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *arXiv preprint arXiv:1708.03949*, 2017.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325, 2016.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.

Elad Hazan and Satyen Kale. Projection-free online learning. In *ICML*, pages 1843–1850, 2012.

Elad Hazan and Kfir Levy. Bandit convex optimization: Towards tight bounds. In *NIPS*, pages 784–792, 2014.

Elad Hazan and Yuanzhi Li. An optimal algorithm for bandit convex optimization. *arXiv preprint arXiv:1603.04350*, 2016.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, pages 427–435, 2013.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Zohar Karnin and Elad Hazan. Hard-margin active linear regression. In *ICML*, pages 883–891, 2014.

Jyrki Kivinen and Manfred K Warmuth. Relative loss bounds for multidimensional regression problems. In *NIPS*, pages 287–293, 1998.

Robert D Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *NIPS*, pages 697–704, 2005.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Peter McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2):179–184, 1970.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2003.

Ankan Saha and Ambuj Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *AISTATS*, pages 636–642, 2011.

Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *COLT*, pages 3–24, 2013.

Yang Yu, Hong Qian, and Yi-Qi Hu. Derivative-free optimization via classification. In *AAAI*, pages 2286–2292, 2016.

Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven C. H. Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *ICML*, pages 4054–4062, 2017.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.