Projection Free Online Learning over Smooth Sets

Kfir Y. Levy
ETH Zurich

Andreas Krause
ETH Zurich

Abstract

The projection operation is a crucial step in applying Online Gradient Descent (OGD) and its stochastic version SGD. Unfortunately, in some cases, projection is computationally demanding and inhibits us from applying OGD. In this work, we focus on the special case where the constraint set is smooth and we have an access to gradient and value oracles of the constraint function. Under these assumptions we design a new approximate projection operation that necessitates only logarithmically many calls to these oracles. We further show that combining OGD with this new approximate projection, results in a projection-free variant that recovers the standard rates of the fully projected version. This applies to both convex and strongly-convex online settings.

1 Introduction

Over the last decade, the Online Gradient Descent (OGD) method introduced by Zinkevich (2003) and its stochastic version SGD have become the methods of choice both in practical machine learning tasks as well as in theory. Usually, OGD/SGD is constrained to choosing points among a set $\mathcal{K}$. This induces simple solutions (e.g., solutions with low norm or low rank), and enables to establish generalization bounds for SGD (Shalev-Shwartz et al., 2009; Cesa-Bianchi et al., 2004). Therefore, in every step, OGD/SGD apply a *projection step* onto the constraint set $\mathcal{K}$. In this paper, we adopt the setup of Mahdavi et al. (2012), and focus on constraints of the following form,

$$\mathcal{K} := \{x \in \mathbb{R}^d : \ h(x) \leq 0\} \ , \tag{1}$$

where $h : \mathbb{R}^d \mapsto \mathbb{R}$ is a smooth convex function, and assume that we may query the gradients/values of $h(\cdot)$.

The projection operation requires to find a point in $\mathcal{K}$ which is closest in ℓ_2 norm to a given point outside $\mathcal{K}$. This translates to a quadratic optimization problem over $\mathcal{K}$. Unfortunately, in some situations, this problem is computationally demanding and even impractical. One relevant example is the case when we have a quadratic constraint of the form $h(x) = x^\top Ax - b$, where $x \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$, and $b \in \mathbb{R}$ (A is a PSD matrix). A projection in this case requires to factorize the matrix A which necessitates $O(d^3)$ operations. This cost is unacceptable in large scale scenarios.

There are roughly two approaches to avoid using projections. The first approach is originally related to Frank and Wolfe (1956). Such FW methods assume an access to a linear optimization oracle over $\mathcal{K}$. There is a large body of work on how to employ such oracles in order to ensure convergence in the stochastic and online optimization settings (Hazan and Kale, 2012; Garber and Hazan, 2013; Lan and Zhou, 2016). On the downside, a linear oracle is not always easy to compute. Concretely, in the case of quadratic constraints, linear optimization might be as time consuming as full projection. Moreover, the online FW version (Hazan and Kale, 2012) obtains suboptimal regret guarantees compared to projected OGD[1].

The second approach, which was originally suggested in Mahdavi et al. (2012) is to solve a primal dual problem that is related to the original optimization problem, while applying only a single/few projections. This work assumes that $\mathcal{K}$ is of the form of Eq. (1)[2], and requires a single query to the gradients of $h(\cdot)$ in each round. On the downside, it does not apply to the online learning setting, but rather to the stochastic and offline optimization settings. Moreover, this approach still

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

[1]Note that in the case of polytope constraints it was shown in Garber and Hazan (2013) how to achieve the same regret bounds as of projected OGD while using a FW style procedure. Nevertheless, their result does not capture smooth sets.

[2]The approach of Mahdavi et al. (2012) actually allows $h(\cdot)$ to be non-smooth.

requires to compute a single/few projections, which might be prohibitive in large scale problems with, e.g., quadratic constraints.

Contribution: We present a different approach towards projection free online convex optimization problems. We assume a constraint $\mathcal{K}$ in the form of Eq. (1), and devise a method that recovers the standard regret rates of $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\log T)$ in the settings of online convex (Zinkevich, 2003) and strongly-convex (Hazan et al., 2007) optimization. Our approach does not require any projections, but rather requires logarithmically many queries to the gradients and values of $h(\cdot)$ in each round.

Concretely, in the case of stochastic/online optimization with quadratic constraints, i.e., $h(x) = x^\top Ax - b$; our method has a computational cost of $O(d^2 \log(T)T)$ over all T rounds (related to the cost of computing values and gradients of $h(\cdot)$). Conversely, all other approaches, including full projections, FW (Hazan and Kale, 2012) and (Mahdavi et al., 2012) require $\mathcal{O}(d^3 + d^2T)$ operations, since they need to factorize A.

At the heart of our method is a new approximate projection procedure which we call FAstProj. This procedure is an efficient substitute to the full projection, and is guided by a simple geometric intuition. Note that the regret bounds that we recover depend on a *curvature* parameter, which encapsulates the geometry of $\mathcal{K}$. We further comment that our results immediately translate into stochastic optimization guarantees by standard online to batch conversion, (Cesa-Bianchi et al., 2004).

Related work: Kalai and Vempala (2005) were the first to provide a projection free online learning method. Similarly to FW, their algorithm employs a linear optimization oracle which they combine with an appropriate noise perturbation. Nevertheless, their results are restricted to linear loss functions. The general convex case was tackled by Hazan and Kale (2012), which offered an online FW variant, though with suboptimal regret guarantees. Later, for the case of polytope constraints, Garber and Hazan (2013) offered a FW variant that achieves the optimal regret rates.

Lan and Zhou (2016) and Lan et al. (2017) developed a FW method that obtains the optimal rates in the convex and strongly-convex stochastic optimization settings. Their ideas were further developed in Hazan and Luo (2016) who devised a variance reduced FW versions. In the broader context of machine learning and optimization, we have recently witnessed an extensive investigation of FW methods. This was done, e.g., in Jaggi (2013); Lacoste-Julien et al. (2013); Garber and Hazan (2015); Lacoste-Julien and Jaggi (2015); Garber and Meshi (2016); Garber (2016); Allen-Zhu et al. (2017), among other works.

Mahdavi et al. (2012) discuss the stochastic convex setting and show how to obtain the standard convergence rates using only single/few projections. Conversely to FW methods, Mahdavi et al. (2012) do not assume the availability of a linear optimization oracle. Instead they assume that $\mathcal{K}$ is given in the form of Eq. (1), and query the gradients of $h(\cdot)$ in each round. Their technique relies on primal-dual machinery. This idea was later developed by Cotter et al. (2016); Chen et al. (2016); Yang et al. (2017) to tackle the cases of large number of constraints as well as to the offline setting. However, all of these works: **(i)** require at least one projection, and **(ii)** are inappropriate to handle the online setting.

Finally, we remark that interior point methods are appropriate to solving constrained convex problems without projecting, (Nesterov and Nemirovskii, 1994). Nevertheless, these methods are too costly in high dimensions. Interestingly, Abernethy et al. (2012) combine interior point mechanism in solving online learning problems.

2 Setting and Preliminaries

Notation: $\|\cdot\|$ denotes the ℓ_2 norm, and $[T] := \{1, \ldots, T\}$. Given two vectors $x, y \in \mathbb{R}^d$ then $[x, y]$ denotes the line segment between them. For a set $\mathcal{K} \subset \mathbb{R}^d$ its diameter is defined as $D = \sup_{x,y\in\mathcal{K}} \|x-y\|$. We also denote by $\partial\mathcal{K}$ and $\mathrm{int}(\mathcal{K})$ the boundary and interior of $\mathcal{K}$.

Online Learning: We consider a repeated game of T rounds between a player and an adversary, at each round $t \in [T]$,

1. player chooses a point $x_t \in \mathcal{K}$.
2. adversary chooses a loss function $f_t : \mathcal{K} \mapsto \mathbb{R}$.
3. player suffers a loss $f_t(x_t)$ and receives $f_t(\cdot)$ as a feedback.

In the OCO (Online Convex Optimization) framework we assume that the decision set $\mathcal{K}$ is convex and that all loss functions are convex. We will also discuss the case where losses are strongly-convex.

We measure the performance of the player using the regret which is the difference between the cumulative loss of the player and the cumulative loss of the best point in hindsight,

$$\mathrm{Regret}_T := \sum_{t=1}^{T} f_t(x_t) - \min_{w^*\in\mathcal{K}} \sum_{t=1}^{T} f_t(w^*) \ .$$

The player aims at minimizing her regret, and we are interested in players that ensure $o(T)$ regret for any loss sequence that the adversary may choose. In this paper we focus on first order online methods, i.e., methods which only require to query the gradients of the loss functions as a feedback.

Projected OGD: The projected OGD algorithm is a well known method in online learning. The update rule of OGD is of the following form,

$$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t))$$

where,

$$\forall z \in \mathbb{R}^d;\ \Pi_{\mathcal{K}}(z) := \arg\min_{y \in \mathcal{K}} \|y - z\| \ .$$

Thus, in each round OGD updates its predictions in the direction opposite to the gradient $\nabla f_t(x_t)$, and then projects onto $\mathcal{K}$. Later, we will present online methods of the same form which utilize a cheap projection procedure instead of $\Pi_{\mathcal{K}}(\cdot)$.

Smooth sets: As previously mentioned we focus on convex compact sets of the following form,

$$\mathcal{K} := \{x \in \mathbb{R}^d :\ h(x) \leq 0\} \ . \tag{2}$$

where $h : \mathbb{R}^d \mapsto \mathbb{R}$ is a smooth convex function. In case $\mathcal{K}$ is of the above form and $h(\cdot)$ is β_h-smooth we will relate to $\mathcal{K}$ as a *β_h-smooth set*. Note that whenever we relate to $\mathcal{K}$ in this paper we assume it has a from as in Eq. (2) (we will not always mention $h(\cdot)$, and relate to $\mathcal{K}$ and $h(\cdot)$ interchangeably). We assume to have an access for both value and gradient oracles of $h(\cdot)$, i.e., we may query the value and gradient of h at any point $x \in \mathbb{R}^d$.

Next we define H-strongly-convex and β-smooth functions. A function $f : \mathcal{K} \mapsto \mathbb{R}$ is *H-strongly convex* over $\mathcal{K}$ if $\forall x, y \in \mathcal{K}$,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{H}{2}\|x - y\|^2 \ .$$

A function $f : \mathcal{K} \mapsto \mathbb{R}$ is *β smooth* over $\mathcal{K}$ if $\forall x, y \in \mathcal{K}$,

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\beta}{2}\|x - y\|^2 \ .$$

Preliminaries:

Definition 2.1. *Given a closed convex set $\mathcal{K}$ and $x \in \partial\mathcal{K}$, n is a* normal vector to $\mathcal{K}$ *in x if,*

$$n \cdot (y - x) \leq 0; \quad \forall y \in \mathcal{K} \ .$$

If, in addition, $\|n\| = 1$ we say that n is a unit normal vector.

The following lemma characterizes the normal vectors of convex sets in the form of Eq. 2,

Lemma 2.1. *Let $\mathcal{K}$ be a convex set (Eq. (2)) and $x \in \partial\mathcal{K}$, then $\nabla h(x)$ is a normal vector to $\mathcal{K}$ in x.*

Given a β_h-smooth convex and compact set $\mathcal{K} = \{x \in \mathbb{R}^d : h(x) \leq 0\}$, we shall now define a geometric quantity that measures how "pointy" is $\mathcal{K}$,

Definition 2.2 (Curvature)**.** *Given a β_h-smooth convex and compact set $\mathcal{K}$ and $x \in \partial\mathcal{K}$ we define the* Local Curvature (LC) *of $\mathcal{K}$ in x as follows,*

$$\mu_x := \beta_h / \|\nabla h(x)\| \ .$$

We also define the Global Curvature (GC) *of $\mathcal{K}$ as follows, $\mu := \max_{x \in \partial\mathcal{K}} \mu_x$.* [3]

Intuitively, if the gradients on the boundary are very small, then the shape of $\partial\mathcal{K}$ will be more "pointy" and the curvature μ will be larger. Note that the above definition is not necessarily equivalent to the definition of curvature from differential geometry. Interestingly, both definitions coincide in the case where $\mathcal{K}$ is the ℓ_2 ball (i.e., $h(x) = \|x\|^2 - R^2$). In this case, $\mu = 0.5/R$, where R is the radius.

In this paper, we assume that μ is upper bounded. This assumption is equivalent to the assumptions made by Mahdavi et al. (2012); Chen et al. (2016); Yang et al. (2017). Concretely, in these works it is assumed that $\min_{x \in \partial\mathcal{K}} \|\nabla h(x)\| \geq B$ which immediately translates to an upper bound on μ, i.e., $\mu \leq \beta_h / B$.

3 Fast Approximate Projection

Here we introduce our FAstProj (Fast Approximate Projection) procedure. This algorithm requires logarithmically many calls to a value and gradient oracles for $h(\cdot)$ in order to provide an approximate projection onto $\mathcal{K}$.

Algorithm description: Our FAstProj procedure is given in Alg. 1. Let us first describe it using Figure 1. Given $x \in \mathcal{K}, v \in \mathbb{R}^d$ this procedure finds an approximate projected point, $\tilde{\Pi}_{\mathcal{K}}(x + v) \in \mathcal{K}$.

In the case where $x + v$ belongs to $\mathcal{K}$ our method returns $x + v$. The interesting case is when $x + v \notin \mathcal{K}$. In this case, our method first finds $\tilde{x}$, which is a point where the segment $(x, x + v]$ intersects with $\partial\mathcal{K}$ (see Fig. 1), and we also define $\tilde{v} := x + v - \tilde{x}$. At this point, our method computes the unit normal vector to $\mathcal{K}$ in $\tilde{x}$, n (represented by the orange dashed line)

[3] Note that given a small enough $\varepsilon > 0$ we can similarly define, $\mu^{(\varepsilon)} := \max_{x : h(x) \in [-\varepsilon, 0]} \beta_h / \|\nabla h(x)\|$. For $\varepsilon = 0$ this coincides with the above definition.

Algorithm 1 Fast Approximate Projection (FAstProj)

Input: $x \in \mathcal{K}$, $v \in \mathbb{R}^d$
Output: approximate projection of $x+v$ onto $\mathcal{K}$
If $x+v \in \mathcal{K}$, **then** output $\tilde{\Pi}_{\mathcal{K}}(x+v) = x+v$
Else
Find $\tilde{x} \in (x, x+v] \cap \partial\mathcal{K}$, and let $\tilde{v} \leftarrow x+v-\tilde{x}$
(find $O(\|v\|^2)$-approximately using bisection)
Calculate $n = \nabla h(\tilde{x})/\|\nabla h(\tilde{x})\|$
Let $\hat{v} \leftarrow \tilde{v} - (n \cdot \tilde{v})n$
Compute $\alpha = \min\{\alpha \geq 0 : \ \tilde{x} + \hat{v} - \alpha n \in \mathcal{K}\}$
(find $O(\|v\|^2)$-approximately using bisection)
Output: $\tilde{\Pi}_{\mathcal{K}}(x+v) = \tilde{x} + \hat{v} - \alpha n$
EndIf

and computes $\hat{v}$ (the dashed blue arrow) which is the projection of $\tilde{v}$ onto the linear subspace orthogonal to n. Next our method finds the first point where the ray $\{\tilde{x} + \hat{v} - \alpha n : \alpha \geq 0\}$ intersects with $\mathcal{K}$ and outputs this point as the approximated projection $\tilde{\Pi}_{\mathcal{K}}(x+v)$ (the red point in the figure).

Let us provide some intuition behind our method. Recall the full projection operation $\Pi(x+v)$ will find a point in $\mathcal{K}$ which is closer to $x+v$ rather than any other point in $\mathcal{K}$. Our approximate procedure tries to do so efficiently while suffering an additional approximate term. Concretely, note that since n is the normal vector to $\mathcal{K}$ in $\tilde{x}$, intuitively the point $\tilde{x}+\hat{v}$ is closer to $x+v$ rather than any other point in $\mathcal{K}$, i.e.,

$$\|(x+v) - (\tilde{x}+\hat{v})\| \leq \|(x+v) - y\|; \quad \forall y \in \mathcal{K} \ .$$

which is exactly what the full projection is trying to obtain. Note however, that the point $\tilde{x}+\hat{v}$ is not necessarily in K. Fortunately by starting at $\tilde{x}+\hat{v}$, we can search in the direction opposite to n and find a point $\tilde{\Pi}_{\mathcal{K}}(x+v) \in \mathcal{K}$ that is close enough to $\tilde{x}+\hat{v}$. By close enough, we mean that for smooth sets the following applies,

$$\|\tilde{\Pi}_{\mathcal{K}}(x+v) - (\tilde{x}+\hat{v})\| \leq \mu\|v\|^2 \ .$$

Using the triangle inequality, we conclude that,

$$\|(x+v) - \tilde{\Pi}_{\mathcal{K}}(x+v)\| \leq \|(x+v)-y\| + \mu\|v\|^2; \quad \forall y \in \mathcal{K} \ .$$

So compared to the full projection we suffer an additional additive term of $\mu\|v\|^2$.

Now, consider a gradient update rule with approximate projection, $x_{t+1} := \tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t \nabla f(x_t))$. In this case the additive term is $\mu\|v\|^2 := \mu\eta_t^2\|\nabla f(x_t)\|^2$, and as we show later on, this term still enables to extract the standard regret guarantees in the convex and strongly-convex online settings (while suffering larger constants which depend on the geometry of $\mathcal{K}$).

The next theorem provides us with the formal guarantees of FAstProj (Alg. 1).

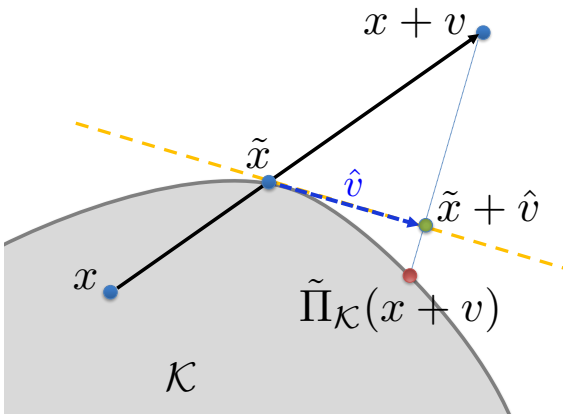

Figure 1: Approximate Projection.

Theorem 3.1. *Let $\mathcal{K}$ be a β_h-smooth and convex set with a Global Curvature of μ, and diameter D. Then upon invoking* FAstProj *(Alg. 1) with $x \in \mathcal{K}$ and $v \in \mathbb{R}^d$ such that $\|v\| \leq 0.5/\mu$ it outputs a point $\tilde{\Pi}_{\mathcal{K}}(x+v) \in \mathcal{K}$, such that the following holds,*

$$\forall y \in \mathcal{K}, \ \|\tilde{\Pi}_{\mathcal{K}}(x+v) - y\| \leq \|(x+v) - y\| + \mu\|v\|^2 \tag{3}$$

Algorithm 1 requires $O(\log(1/\|v\|)) + \log(1+\|v\|))$ calls to a value and gradient oracle for $h(\cdot)$.

Efficient implementation: Clearly, we can not compute the exact values $\tilde{x}$ and α. Instead, we compute these values up to sufficient accuracy. Next we describe how to achieve an accuracy of ε with only $\log(1/\varepsilon)$ calls to gradient and value oracles for $h(\cdot)$.

First, let us focus on finding $\tilde{x}$, which is equivalent to finding an arbitrary root of the equation $h(x) = 0$ in the segment $(x, x+v]$. Since $h(x) \leq 0$ and $h(x+v) > 0$, we can use bisection in order to find a point which is ε close the exact value of $\tilde{x}$ using $O(\log(1/\varepsilon))$ calls to the value of $h(\cdot)$ (note that even if $h(x) = 0$ we can still use bisection, we elaborate on this in the full proof of the theorem).

In order to find α, assume for now that we are given a point $z \in \mathcal{K} \cap \{\tilde{x} + \hat{v} - \alpha n : \alpha \geq 0\}$. In this case, clearly $h(z) \leq 0$ and $h(\tilde{x}+\hat{v}) > 0$, and both points $z, \tilde{x}+\hat{v}$, are on the ray $\{\tilde{x} + \hat{v} - \alpha n : \alpha \geq 0\}$. Therefore, we can again use bisection in order to find a point ε close to the optimal α using $O(\log(1/\varepsilon))$ calls to the value of $h(\cdot)$. The question now is: *how to come up with a point $z \in \mathcal{K} \cap \{\tilde{x} + \hat{v} - \alpha n : \alpha \geq 0\}$?* In the full proof of the theorem, we show how the problem of finding such z translates to approximately solving a one dimensional convex optimization problem. The latter can be done by convex bisection[4] (Juditsky, 2015), which requires $O(\log(1+\|v\|))$ calls to a gradient oracle of $h(\cdot)$.

[4]Convex bisection is, in a sense, the one-dimensional

Taking an accuracy parameter of $\varepsilon \leq \mathcal{O}(\|v\|^2)$ is sufficient in order to maintain the guarantees of Theorem 3.1 (although with slightly worse constants).

3.1 Proof of Theorem 3.1

Here we provide a proof sketch of Theorem 3.1. First, let us introduce our main technical lemma,

Lemma 3.1. *Let $\mathcal{K}$ be a β_h-smooth and convex set with a Global Curvature of μ, and diameter D. Also let $x \in \partial\mathcal{K}$ and $n_x := \nabla h(x)/\|\nabla h(x)\|$, and let $v \in \mathbb{R}^d$ such that $v \cdot n_x = 0$, and $\|v\| \leq 0.5/\mu$. Then there exists $0 \leq \alpha \leq \mu_x\|v\|^2 \leq \mu\|v\|^2$ such that $x + v - \alpha n_x \in \mathcal{K}$. Moreover, for any $\Delta_\alpha \in [0, \frac{4}{5\mu}]$, the point $x + v - (\alpha + \Delta_\alpha)n_x \in \mathcal{K}$.*

Proof Sketch of Theorem 3.1. The Theorem is immediate in case that $x + v \in \mathcal{K}$ since then $\tilde{\Pi}_{\mathcal{K}}(x + v) = x + v \in \mathcal{K}$. Next we focus on the case where $x + v \notin \mathcal{K}$.

In this case $\tilde{x} \in \partial\mathcal{K}$ with a normal vector n and $\hat{v}$ is orthogonal to n (by definition of $\hat{v}$). In addition, $\|\hat{v}\| \leq \|v\| \leq 0.5/\mu$, and we can therefore apply Lemma 3.1 (taking $x \leftrightarrow \tilde{x}$ and $v \leftrightarrow \hat{v}$), implying,

$$\|\tilde{\Pi}_{\mathcal{K}}(x + v) - (\tilde{x} + \hat{v})\| = \alpha \leq \mu\|\hat{v}\|^2 \leq \mu\|v\|^2 \ . \quad (4)$$

where we have used $\tilde{\Pi}_{\mathcal{K}}(x+v) := \tilde{x} + \hat{v} - \alpha n_x$ together with $\|n_x\| = 1$. We also used the definition of α in Alg. 1, which together with Lemma 3.1 implies that $\alpha \leq \mu\|\hat{v}\|^2$.

Next notice that $x+v = \tilde{x}+\tilde{v}$, and that the dependence of FAstProj (Alg. 1) in $\tilde{v}$ is only through $\hat{v}$ (which is the projection of $\tilde{v}$ onto the linear space orthogonal to n). Thus, the following holds,

$$\tilde{\Pi}_{\mathcal{K}}(x + v) = \tilde{\Pi}_{\mathcal{K}}(\tilde{x} + \tilde{v}) = \tilde{\Pi}_{\mathcal{K}}(\tilde{x} + \hat{v}) \ . \quad (5)$$

Also, using the definition of $\tilde{x}, n$, as well as the the smoothness of $\mathcal{K}$ it can be shown that (see Fig. 1),

$$\forall y \in \mathcal{K},\ \|(\tilde{x} + \hat{v}) - y\| \leq \|(\tilde{x} + \tilde{v}) - y\| = \|(x + v) - y\| \ . \quad (6)$$

Combining Equations (4), (5), (6), implies that $\forall y \in \mathcal{K}$,

$$\begin{aligned}\|\tilde{\Pi}_{\mathcal{K}}(x + v) - y\| &= \|\tilde{\Pi}_{\mathcal{K}}(\tilde{x} + \hat{v}) - y\| \\ &\leq \|(\tilde{x} + \hat{v}) - y\| + \|(\tilde{x} + \hat{v}) - \tilde{\Pi}_{\mathcal{K}}(\tilde{x} + \hat{v})\| \\ &\leq \|(x + v) - y\| + \mu\|v\|^2 \ .\end{aligned}$$

□

version of the ellipsoid method. The idea is that in the 1 dimensional convex case the sign of the gradient at a point x "tells" us whether the global optimum is larger or smaller than x. Thus, using gradient information we can use bisection in order to approximately find the optimum.

3.1.1 Proof Sketch of Lemma 3.1

Next we provide a proof sketch of Lemma 3.1. To simplify the exposition we will only sketch a part of the proof (see the appendix for the full details).

Proof Sketch of Lemma 3.1. Lemma 3.1 is a direct consequence of the following three lemmas.

Lemma 3.2. *Under the same assumptions of Lemma 3.1. If $\|v\| \leq 0.5/\mu$, and if there exists $\alpha \geq 0$ such that $x + v - \alpha n_x \in \partial\mathcal{K}$ then the following holds,*

$$\alpha \leq \mu_x\|v\|^2 \leq \mu\|v\|^2 \ . \quad (7)$$

where $\mu_x := \beta_h/\|\nabla h(x)\|$ is the Local Curvature at x (see Def. 2.2).

Lemma 3.3. *Under the same assumptions of Lemma 3.1. If $\|v\| \leq 0.5/\mu$, then there exists $\alpha \geq 0$ such that $x + v - \alpha n_x \in \partial\mathcal{K}$.*

Lemma 3.4. *Under the same assumptions of Lemma 3.1. If $\|v\| \leq 0.5/\mu$, then there exists $\alpha \geq 0$ such that $x + v - (\alpha + \Delta_\alpha)n_x \in \mathcal{K}$ for any $\Delta_\alpha \in [0, \frac{4}{5\mu}]$.*

In the appendix, we prove Lemma 3.2 and then use it in order to prove Lemma 3.3; both are then used to prove Lemma 3.4.

Next we provide a proof sketch for Lemma 3.2. Concretely, we will show that if there exists $\alpha \geq 0$ such that $x + v - \alpha n_x \in \partial\mathcal{K}$ then the following holds,

$$\alpha \leq \mu_x\|v\|^2 \leq \mu\|v\|^2 \ . \quad (8)$$

Indeed let us denote $z := x + v - \alpha n_x$ and assume $z \in \partial\mathcal{K}$. Using the β_h smoothness of $h(\cdot)$ yields,

$$\begin{aligned}0 = h(z) - h(x) &\leq \nabla h(x) \cdot (z - x) + \frac{\beta_h}{2}\|x - z\|^2 \\ &= -\alpha\|\nabla h(x)\| + \frac{\beta_h}{2}\left(\|v\|^2 + \alpha^2\right) \quad (9)\end{aligned}$$

where we use $h(x) = h(z) = 0$ (since $x, z \in \partial\mathcal{K}$), as well as, $n_x := \nabla h(x)/\|\nabla h(x)\|$, and $v \cdot n_x = v \cdot \nabla h(x) = 0$.

Now Eq. (9) is a quadratic inequality in α, and is equivalent to the following,

$$\alpha \leq \frac{\|\nabla h(x)\|}{\beta_h}\left(1 - \sqrt{1 - \frac{\beta_h^2\|v\|^2}{\|\nabla h(x)\|^2}}\right) \quad (10)$$

Using $\|v\| \leq 0.5/\mu$ and recalling the definitions of μ_x, μ allows to derive Eq. (8) from Eq. (10). □

4 Regret Guarantees

Here we establish the guarantees of the online gradient descent (OGD) method when we apply FAstProj rather than the full projection. As we show in Theorems 4.1, and 4.2, this OGD version preserves the well known regret rates of $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\log T)$ in the convex and strongly-convex settings. Our bounds are worse by a multiplicative factor compared to projected OGD.

4.1 Convergence Guarantees

In this section, we assume the online convex optimization setting (see Section 2), where we receive a a sequence of convex functions loss $\{f_t : \mathcal{K} \mapsto \mathbb{R}\}_{t\in[T]}$. We also assume bounded gradients, i.e., $\forall t \in [T], x \in \mathcal{K};\ \|\nabla f_t(x)\| \le G$.

Consider the following OGD version combined with FAstProj (Alg. 1) rather than exact projection,

$$\forall t \in [T]; \quad x_{t+1} = \tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t \nabla g_t) \ . \tag{11}$$

where $g_t := \nabla f(x_t)$.

Recall the definition of regret,

$$\text{Regret}_T := \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \ .$$

The following two theorems establish the regret guarantees of this algorithm in the convex and strongly-convex cases.

Theorem 4.1. *Assume that $\mathcal{K}$ is a β_h smooth convex set with bounded diameter D. Upon invoking the OGD algorithm with FAstProj (Eq. (11)) and taking an arbitrary $x_1 \in \mathcal{K}$, and $\eta_t = \frac{D}{G(1+2\mu D)\sqrt{t}}$, the following holds,*

$$\textit{Regret}_T \le \mathcal{O}\left((1+\mu D) G D \sqrt{T}\right) \ .$$

The next theorem asserts the guarantees in case that all loss function $\{f_t\}_{t\in[T]}$ are H-strongly-convex.

Theorem 4.2. *Assume that $\mathcal{K}$ is a β_h smooth convex set with bounded diameter D. Upon invoking the OGD algorithm with FAstProj (Eq. (11)) and taking an arbitrary $x_1 \in \mathcal{K}$, and $\eta_t = (2\mu G + Ht)^{-1}$, the following holds,*

$$\textit{Regret}_T \le \mathcal{O}\left(\frac{G^2(1+\mu D)}{H} \log T\right) \ .$$

Notice that in both cases our bounds are worse by a multiplicative factor of $\mathcal{O}(\mu D)$ compared to projected OGD.

Invariance of μD: Before we go on to the proofs we show that the μD factor is invariant with respect to a very natural quantity. Concretely, consider convex constraints of the form $\mathcal{K}_C = \{x \in \mathbb{R}^d : c(x) \le C\}$ where $c(\cdot)$ is a *homogeneous of degree* 2 smooth convex function, and $C > 0$.

Proposition 4.1. *As long as $\mathcal{K}_C$ is non-empty then the factor μD of $\mathcal{K}_C$ is invariant w.r.t. C.*

By *homogeneous of degree* 2 we mean that $\forall \rho \in \mathbb{R};\ c(\rho x) = \rho^2 c(x)$. Note that for such functions, it can be shown that $\nabla c(\rho x) = \rho \nabla c(x)$. In order to see why the proposition holds, consider $\mathcal{K}_C$ and $\mathcal{K}_{\rho C}$ for some $\rho > 0$. Due to homogeneity of degree 2, the boundary of $\mathcal{K}_{\rho C}$ is a $\sqrt{\rho}$ scaled version of the boundary of $\mathcal{K}_C$. Thus, $\text{diameter}(\mathcal{K}_{\rho C}) = \sqrt{\rho} \cdot \text{diameter}(\mathcal{K}_{\rho C})$, and also, $\min_{x\in\partial\mathcal{K}_{\rho C}} \|\nabla c(x)\| = \sqrt{\rho} \cdot \min_{x\in\partial\mathcal{K}_C} \|\nabla c(x)\|$. Since the smoothness of the body $\mathcal{K}_C$ (equivalently smoothness of $c(x) - C$) does not change C, this implies that the μD factor for $\mathcal{K}_C$ and $\mathcal{K}_{\rho C}$ is the same.

The above assumptions on $c(\cdot)$ hold for quadratic constraints, i.e., $c(x) = x^\top A x$, as well as for squared ℓ_p norms, i.e., $c(x) = \|x\|_p^2$ with $p \in (1, \infty)$.

4.2 Applications: Quadratic Constraints

Consider a quadratic constraint: $h(x) = \|Ax - y\|^2 - b$, where $x \in \mathbb{R}^d, y \in \mathbb{R}^m, b \in \mathbb{R}$, and $A \in \mathbb{R}^{m\times d}$. Such constraints are prevalent in optimization and machine learning. Concrete examples include training Kernelized SVMs (where $h(x) = x^\top K x - b$, and K is the kernel matrix), and compressive sensing (Candès and Wakin, 2008).

Next, we show that for quadratic constraints, the curvature μ is upper bounded and we can therefore apply our approach. Clearly, the smoothness of $h(\cdot)$ is equal to the largest eigenvalue of $A^\top A$, i.e., $\beta_h = \lambda_{\max}(A^\top A)$. Also, as Yang et al. (2017) shows $\min_{x:h(x)=0} \|\nabla h(x)\| \ge \sqrt{b\lambda_{\min}(AA^\top)}$. This immediately implies that $\mu = \lambda_{\max}(A^\top A)/\sqrt{b\lambda_{\min}(AA^\top)}$, and $\mu D \le \lambda_{\max}(A^\top A)/\sqrt{\lambda_{\min}(AA^\top)}$.

4.3 Proof of Theorem 4.1

Proof of Theorem 4.1. First note that due to the choice of learning rate then for any $t \in [T]$,

$$\|\eta_t g_t\| \le \frac{D}{2\mu D} \cdot \frac{\|g_t\|}{G\sqrt{t}} \le 0.5/\mu \ .$$

The above enables to apply Theorem 3.1 which implies that $\forall x \in \mathcal{K}$,

$$\|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\| \le \|(x_t - \eta_t g_t) - x\| + \mu\eta_t^2 \|g_t\|^2$$

Taking the square of the above we get,

$$\begin{aligned}
&\|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\|^2 \qquad (12)\\
&\leq \|(x_t - \eta_t g_t) - x\|^2 + \mu^2\eta_t^4\|g_t\|^4 \\
&\quad + 2\|(x_t - \eta_t g_t) - x\| \cdot \mu\eta_t^2\|g_t\|^2 \\
&\leq \|(x_t - \eta_t g_t) - x\|^2 + \mu^2\eta_t^4 G^4 + 2(D + \eta_t G)\mu\eta_t^2 G^2 \\
&\leq \|(x_t - \eta_t g_t) - x\|^2 + \mu^2\eta_t^4 G^4 + (2\mu D + 1)\eta_t^2 G^2 \qquad (13)
\end{aligned}$$

where we used $\|g_t\| \leq G$, $\|x_t - x\| \leq D$ and $\eta_t G \leq 0.5/\mu$.

We continue similarly to the classical OGD proof. For any $t \in [T]$ we may use the above to show the following for any $x \in \mathcal{K}$,

$$\begin{aligned}
&\|x_{t+1} - x\|^2 \\
&= \|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\|^2 \\
&\leq \|(x_t - \eta_t g_t) - x\|^2 + \mu^2\eta_t^4 G^4 + (2\mu D + 1)\eta_t^2 G^2 \\
&\leq \|x_t - x\|^2 - 2\eta_t g_t \cdot (x_t - x) + \eta_t^2\|g_t\|^2 \\
&\quad + \mu^2\eta_t^4 G^4 + (2\mu D + 1)\eta_t^2 G^2
\end{aligned}$$

Re-arranging the above and using $\|g_t\| \leq G$ we get,

$$\begin{aligned}
g_t \cdot (x_t - x) \;\leq\; &\frac{1}{2\eta_t}\left(\|x_t - x\|^2 - \|x_{t+1} - x\|^2\right) \\
&+ \frac{1}{2}\mu^2\eta_t^3 G^4 \quad + \eta_t(\mu D + 1)G^2
\end{aligned}$$

By convexity $f_t(x_t) - f_t(x) \leq \nabla f(x_t) \cdot (x_t - x)$. Using this and summing the above inequality over all rounds we conclude that $\forall x \in \mathcal{K}$,

$$\begin{aligned}
&\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x) \\
&\quad \leq \sum_{t=1}^T \frac{\|x_t - x\|^2}{2}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + G^2(1 + \mu D)\sum_{t=1}^T \eta_t \\
&\qquad + \frac{G^4\mu^2}{2}\sum_{t=1}^T \eta_t^3 \\
&\quad \leq \frac{D^2}{2\eta_T} + G^2(1 + \mu D)\eta_1 \sum_{t=1}^T \frac{1}{\sqrt{t}} + \frac{G^4\mu^2\eta_1^3}{2}\sum_{t=1}^T \frac{1}{t^{3/2}} \\
&\quad \leq \left((1 + \mu D)/2 + 2\right) GD\sqrt{T} + C \;.
\end{aligned}$$

here in the first line we denote $\eta_0 = \infty$, the second line uses $\mathrm{diam}(\mathcal{K}) = \mathrm{D}$ and $\eta_t \leq \eta_{t-1}$. The last line uses $\sum_{t=1}^T 1/\sqrt{t} \leq 2\sqrt{T}$, and $\sum_{t=1}^T 1/t^{3/2} \leq 3$. Also $C := \frac{3}{2}G\mu^2 D^3/(1 + \mu D)^3$. □

4.4 Proof of Theorem 4.2

Proof. First note that due to the choice of learning rate then for any $t \in [T]$,

$$\|\eta_t g_t\| = \frac{\|g_t\|}{2\mu G} \leq 0.5/\mu \;.$$

The above enables to apply Theorem 3.1 which implies that $\forall x \in \mathcal{K}$,

$$\|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\| \leq \|(x_t - \eta_t g_t) - x\| + \mu\eta_t^2\|g_t\|^2$$

Similarly to the proof of Theorem 4.1 , we take the square of the above which gives,

$$\begin{aligned}
&\|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\|^2 \\
&\quad \leq \|(x_t - \eta_t g_t) - x\|^2 + \mu^2\eta_t^4 G^4 + (2\mu D + 1)\eta_t^2 G^2 \;. \qquad (14)
\end{aligned}$$

Using the above it follows that for any $x \in \mathcal{K}$,

$$\begin{aligned}
\|x_{t+1} - x\|^2 &= \|\tilde{\Pi}_{\mathcal{K}}(x_t - \eta_t g_t) - x\|^2 \\
&\leq \left(\|x_t - x\|^2 - 2\eta_t g_t \cdot (x_t - x) + \eta_t^2 G^2\right) \\
&\quad + \mu^2\eta_t^4 G^4 + (2\mu D + 1)\eta_t^2 G^2 \;.
\end{aligned}$$

Re-arranging we get:

$$\begin{aligned}
g_t \cdot (x_t - x) \leq &\frac{1}{2\eta_t}\left(\|x_t - x\|^2 - \|x_{t+1} - x\|^2\right) \\
&+ \eta_t G^2(1 + \mu D) + \frac{\eta_t^3}{2}\mu^2 G^4
\end{aligned}$$

Combining the above with the H-strong-convexity of f_t's and summing over all rounds we conclude that,

$$\begin{aligned}
&\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x) \\
&\quad \leq \sum_{t=1}^T \frac{\|x_t - x\|^2}{2}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - H\right) \\
&\qquad + G^2(1 + \mu D)\sum_{t=1}^T \eta_t + \mu^2 G^4 \sum_{t=1}^T \frac{\eta_t^3}{2} \\
&\quad \leq \mu D^2 G + \frac{G^2(1 + \mu D)}{H}(1 + \log T) + \frac{2\mu^2 G^4}{H^3} \;.
\end{aligned}$$

where we denote $\eta_0 = \infty$. The last line uses $\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} = H$; $\forall t \geq 2$, and $\|x_t - x\| \leq D$, as well as,

$$\sum_{t=1}^T \eta_t \leq \sum_{t=1}^T 1/Ht \leq (1 + \log T)/H \;,$$

and also,

$$\sum_{t=1}^T \eta_t^3 \leq \sum_{t=1}^T 1/(Ht)^3 \leq 2/H^3 \;.$$

□

4.5 Complexity of FAstProj for OGD

As we discuss in the introduction, FAstProj requires $O(\log(T))$ calls to a gradient+value oracles for $h(\cdot)$ in each round of OGD. Here we explain the reason for that.

Theorem 3.1 tells us that FastProj requires $O(\log(1/\|v\|^2)+\log(1+\|v\|^2))$ calls to a gradient+value oracle at each round. Note that in Theorems 4.1 and 4.2 we take $v = \eta\nabla f_t(x_t)$. Since the gradient magnitude is bounded by G, the complexity term $\log(1 + \|v\|^2)$ is bounded by a constant, i.e., $\log(1 + (\eta G)^2)$. Note however that when $\|v\|$ is very small then the other term might explode. Fortunately, we can ignore rounds where $\|\nabla f_t(x_t)\| \leq 1/T$ and still maintain regret guarantees up to an additive constant factor[5]. Meaning we only have to make gradient update and call FastProj procedure when $\|v\| = \|\eta\nabla f_t(x_t)\| \geq \Omega(\frac{1}{\sqrt{T}T})$. In this case the first term in complexity, $\log(1/\|v\|^2)$, is upper bounded by $O(\log(T))$.

5 Conclusion

We presented a new approach towards projection-free online learning, which recovers the same rates as the fully projected version. While we have mainly focused on projections for first order online methods, second order online methods, e.g., Online Newton Step (Hazan et al., 2007), require more complex projections, which take a data dependent condition matrix into account. It will be interesting to extend our approach to this case.

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No. 815943, as well as from the ETH Zurich Postdoctoral Fellowship and Marie Curie Actions for People COFUND program.

References

J. D. Abernethy, E. Hazan, and A. Rakhlin. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory*, 58(7):4164–4175, 2012.

Z. Allen-Zhu, E. Hazan, W. Hu, and Y. Li. Linear convergence of a frank-wolfe type algorithm over trace-norm balls. In *Advances in Neural Information Processing Systems*, pages 6192–6201, 2017.

E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9): 2050–2057, 2004.

J. Chen, T. Yang, Q. Lin, L. Zhang, and Y. Chang. Optimal stochastic strongly convex optimization with a logarithmic number of projections. 2016.

A. Cotter, M. Gupta, and J. Pfeifer. A light touch for heavily constrained sgd. In *Conference on Learning Theory*, pages 729–771, 2016.

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

D. Garber. Faster projection-free convex optimization over the spectrahedron. In *Advances in Neural Information Processing Systems*, pages 874–882, 2016.

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 420–428. IEEE, 2013.

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

D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Advances in Neural Information Processing Systems*, pages 1001–1009, 2016.

E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1843–1850. Omnipress, 2012.

E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

A. Juditsky. Convex optimization ii: Algorithms. https://ljk.imag.fr/membres/Anatoli.

[5] In the online setting, when the gradients $\|\nabla f_t(x_t)\|$ is smaller than $1/T$, then ignoring the update in these rounds (i.e. not performing any update) only affects the regret by a constant. This means that regret guarantees for the convex and strongly-convex cases remain $O(\sqrt{T})$ and $O(\log T)$.

Iouditski/cours/convex/chapitre_22.pdf, November 2015.

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

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *ICML 2013 International Conference on Machine Learning*, pages 53–61, 2013.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

G. Lan, S. Pokutta, Y. Zhou, and D. Zink. Conditional accelerated lazy stochastic gradient descent. In *International Conference on Machine Learning*, pages 1965–1974, 2017.

M. Mahdavi, T. Yang, R. Jin, S. Zhu, and J. Yi. Stochastic gradient descent with only one projection. In *Advances in Neural Information Processing Systems*, pages 494–502, 2012.

Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *COLT*, 2009.

T. Yang, Q. Lin, and L. Zhang. A richer theory of convex constrained optimization with reduced projections and improved rates. In *International Conference on Machine Learning*, pages 3901–3910, 2017.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.