Distributed Projection-Free Online Learning for Smooth and Convex Losses

Yibo Wang[1,2,†], **Yuanyu Wan**[3,†], **Shimao Zhang**[1], **Lijun Zhang**[1,2,*]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] Peng Cheng Laboratory, Shenzhen 518055, China
[3] School of Software Technology, Zhejiang University, Ningbo 315048, China
{wangyb, zhanglj}@lamda.nju.edu.cn, wanyy@zju.edu.cn, smzhang@smail.nju.edu.cn

Abstract

We investigate the problem of distributed online convex optimization with complicated constraints, in which the projection operation could be the computational bottleneck. To avoid projections, distributed online projection-free methods have been proposed and attain an $\mathcal{O}(T^{3/4})$ regret bound for general convex losses. However, they cannot utilize the smoothness condition, which has been exploited in the centralized setting to improve the regret. In this paper, we propose a new distributed online projection-free method with a tighter regret bound of $\mathcal{O}(T^{2/3})$ for smooth and convex losses. Specifically, we first provide a distributed extension of Follow-the-Perturbed-Leader so that the smoothness can be utilized in the distributed setting. Then, we reduce the computational cost via sampling and blocking techniques. In this way, our method only needs to solve one linear optimization per round on average. Finally, we conduct experiments on benchmark datasets to verify the effectiveness of our proposed method.

Introduction

Distributed online convex optimization (D-OCO) has been a popular research topic due to its powerful capability in distributed online decision making, such as distributed tracking in sensor networks (Li et al. 2002; Shahrampour and Jadbabaie 2018) and distributed energy management (Bao et al. 2018; Lesage-Landry and Callaway 2020). In general, D-OCO can be viewed as an iterative game between an adversary and a group of local learners connected via a distributed network, which is defined by an undirected graph $G = (V, E)$ with the vertex set $V = \{1, \cdots, n\}$ and the edge set $E \subset V \times V$. At round t, learner $i \in V$ chooses a decision $\mathbf{x}_{t,i}$ from a convex set $\mathcal{K} \subseteq \mathbb{R}^d$. After that, the adversary reveals a convex local loss function $f_{t,i}(\cdot) : \mathcal{K} \to \mathbb{R}$ and the local learner i suffers a loss $f_{t,i}(\mathbf{x}_{t,i})$. The main challenge in D-OCO is that each learner i can only access local information (e.g., losses and gradients of itself) but needs to minimize the global loss:

$$f_t(\mathbf{x}) = \sum_{j=1}^{n} f_{t,j}(\mathbf{x}), \tag{1}$$

which is the sum of all local loss functions at round t. Minimizing the global loss is equivalent to minimizing the global *regret*, which is defined as the difference between cumulative global loss of local learner i and that of the best fixed decision in hindsight:

$$\text{Regret}_i = \sum_{t=1}^{T} f_t(\mathbf{x}_{t,i}) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{2}$$

When there is only one learner in the network, this challenge can be handled by a classical online learning framework: centralized online convex optimization (OCO). The global loss f_t in centralized OCO reduces to the local loss for the single learner. Correspondingly, the global *regret* degenerates to that defined by the local loss:

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}), \tag{3}$$

where $\mathbf{x}_t$ is the action of the single learner at round t. In centralized OCO, various methods have been proposed to achieve the optimal regret bound, such as Online Gradient Descent (OGD) (Zinkevich 2003; Hazan, Agarwal, and Kale 2007) and Online Dual Averaging (ODA) (Nesterov 2009; Bubeck 2015). In the past decade, these methods in centralized OCO have been extended to the distributed setting, e.g., Distributed Online Gradient Descent (D-OGD) (Ram, Nedich, and Veeravalli 2010; Yan et al. 2013; Yuan, Ling, and Yin 2016) and Distributed Online Dual Averaging (D-ODA) (Hosseini, Chapman, and Mesbahi 2013; Lee, Nedić, and Raginsky 2016).

A main operation in these D-OCO methods is *projection*, which pulls an infeasible point back into the convex domain $\mathcal{K}$. In general, the infeasible point is projected to the nearest one in domain $\mathcal{K}$ under ℓ_2 distance (Hazan and Kale 2012). However, when constraints are complicated, such an operation can be quite time-consuming and even impossible for the local learners with light computation capability. To handle this issue, recent studies (Zhang et al. 2017; Wan, Tu, and Zhang 2020; Wan et al. 2022) proposed several distributed online projection-free methods, which replace the projection operation with a linear optimization step. Different from the projection operation, the linear optimization step could be more efficient for certain complicated domains. Moreover, existing distributed online projection-free methods can

[†]Contributed equally, [*]Corresponding author

attain an $\mathcal{O}(T^{3/4})$ regret bound for general convex losses (Zhang et al. 2017; Wan, Tu, and Zhang 2020) and a tighter $\tilde{\mathcal{O}}(T^{2/3})$ regret bound for strongly convex losses (Wan et al. 2022).

Besides strong convexity, exploiting the smoothness condition of loss functions is another common approach to promote the regret bound (Srebro, Sridharan, and Tewari 2010; Orabona, Cesa-Bianchi, and Gentile 2012). A recent work (Hazan and Minasyan 2020) proposed the Online Smooth Projection Free algorithm (OSPF) to improve the regret of projection-free methods for smooth losses in centralized OCO. However, in the distributed setting, none of existing research considers the smoothness of loss functions. Inspired by OSPF, we propose Distributed Online Smooth Projection-Free Algorithm (D-OSPA) with a tighter regret bound of $\mathcal{O}(T^{2/3})$ for smooth and convex losses in D-OCO. The main technique contribution of our work is extending Follow-the-Perturbed-Leader (Kalai and Vempala 2005; Hazan 2016) to the distributed setting. Furthermore, to make the distributed extension tractable, we apply sampling and blocking techniques and hence, obtain our final method D-OSPA. Specifically, we first estimate the expectation operation in the distributed extension by computing the average of m i.i.d. samples. Then, we group k rounds into one to reduce the update frequency and therefore, decrease the computational cost. Overall, in D-OSPA, there are m/k linear optimizations per round on average. When $m = k$, D-OSPA only needs to solve one linear optimization in each round averagely. We summarize the contributions of this work as follows:

- We first propose Distributed Follow-the-Perturbed-Leader (D-FPL), which achieves the optimal regret of $\mathcal{O}(\sqrt{T})$ in D-OCO. To the best of our knowledge, D-FPL is the first distributed variant of Follow-the-Perturbed-Leader method.
- Based on D-FPL, we present Distributed Online Smooth Projection-Free Algorithm (D-OSPA) with one linear optimization per round on average. And we prove that for smooth and convex loss functions, D-OSPA ensures an $\mathcal{O}(T^{2/3})$ regret bound.
- Even if loss functions are not smooth, we show that D-OSPA still obtains an $\mathcal{O}(T^{3/4})$ regret guarantee, which matches the result of previous work (Zhang et al. 2017).

Related Work

In this section, we briefly overview existing projection-free methods in centralized online convex optimization (OCO) and distributed online convex optimization (D-OCO).

Projection-Free Methods in Centralized OCO

Hazan and Kale (2012) proposed the first online projection-free method named Online Frank-Wolfe (OFW), which is an online extension of Frank-Wolfe Algorithm (Frank and Wolfe 1956), and ensures an $\mathcal{O}(T^{3/4})$ regret for general convex loss functions. The basic idea is to decrease the computational cost of the projection operation by replacing it with the following linear optimization steps

$$\begin{aligned} \mathbf{v}_t &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \langle \nabla F_t(\mathbf{x}_t), \mathbf{x}\rangle \\ \mathbf{x}_{t+1} &= \mathbf{x}_t + s_t(\mathbf{v}_t - \mathbf{x}_t), \end{aligned} \tag{4}$$

where $F_t(\mathbf{x}) = \eta \sum_{r=1}^{t-1} \nabla f_r(\mathbf{x}_r)^\top \mathbf{x} + \|\mathbf{x} - \mathbf{x}_1\|_2^2$ is a surrogate loss function and η, s_t are two parameters. Based on OFW, several methods are proposed with improved bounds by using additional conditions such as the strong convexity of losses (Wan and Zhang 2021; Kretzu and Garber 2021).

Recently, Hazan and Minasyan (2020) proposed OSPF, which promotes the regret bound from $\mathcal{O}(T^{3/4})$ to $\mathcal{O}(T^{2/3})$ for smooth and convex losses. OSPF is based on a very classical method, Follow-the-Perturbed-Leader (Kalai and Vempala 2005; Hazan 2016) which computes the expected action according to the random variable $\mathbf{v}$ sampled from an unit ball $\mathbb{B}$:

$$\mathbf{x}_t = \mathbb{E}_{\mathbf{v}\sim\mathbb{B}}\left[\operatorname*{argmax}_{\mathbf{x}\in\mathcal{K}} \left\langle -\sum_{r=1}^{t-1} \nabla f_r(\mathbf{x}_r) + \frac{1}{\eta}\cdot\mathbf{v}, \mathbf{x}\right\rangle\right], \tag{5}$$

where $\sum_{r=1}^{t-1} \nabla f_r(\mathbf{x}_r)$ is the sum of historical gradients and η is the perturbation parameter. Note that it is difficult to compute a closed form of the expected action in general. Therefore, OSPF utilizes sampling and blocking techniques to enhance the computational efficiency (Cohen and Hazan 2015; Garber and Kretzu 2020).

Furthermore, we notice that recent work (Garber and Hazan 2016; Levy and Krause 2019; Molinaro 2020; Mhammedi 2022; Garber and Kretzu 2022) improved the regret bound of projection-free methods by exploiting some special conditions of the domain set $\mathcal{K}$.

Projection-Free Methods in D-OCO

Due to the distributed connections, each local learner i can only communicate with its neighbors $N_i = \{j \in V | (i,j) \in E\}$. Following previous studies (Hosseini, Chapman, and Mesbahi 2013; Zhang et al. 2017), we introduce a non-negative weight matrix $P \in \mathbb{R}^{n\times n}$ to model the communication between local learners. By exploiting the communication matrix P, learner i can update actions based on not only its own local information (e.g., losses and gradients) but also that of its neighbors.

More specifically, for each local learner i, we introduce a dual variable $\mathbf{z}_{t,i}$ as an approximation for accumulative global gradients until round t and update $\mathbf{z}_{t,i}$ according to

$$\mathbf{z}_{t,i} = \sum_{j\in N_i} P_{ij}\mathbf{z}_{t-1,j} + \nabla f_{t,i}(\mathbf{x}_{t,i}), \tag{6}$$

where P_{ij} is the weight of the dual variable that learner i receives from learner j and $\nabla f_{t,i}(\mathbf{x}_{t,i})$ is the local gradient of learner i at round t. A common extension for methods in centralized OCO (e.g., ODA and OFW) to the distributed setting is to replace the sum of historical local gradients $\sum_{r=1}^{t-1} \nabla f_{r,i}(\mathbf{x}_{r,i})$ with $\mathbf{z}_{t-1,i}$ and update $\mathbf{z}_{t-1,i}$ to $\mathbf{z}_{t,i}$ according to (6).

In this way, Zhang et al. (2017) proposed a distributed projection-free method, named Distributed Online Conditional Gradient (D-OCG). However, D-OCG requires each learner to share local gradients with their neighbors every round, which leads to an $\mathcal{O}(T)$ communication complexity. To address this issue, Wan, Tu, and Zhang (2020) introduced Distributed Block Online Conditional Gradient (D-BOCG), which reduces the communication complexity to $\mathcal{O}(\sqrt{T})$ and still ensures the same regret. The main idea of D-BOCG is the blocking technique (Garber and Kretzu 2020), which divides the whole rounds into equally-sized blocks and only updates local learners at the beginning of each block.

Main Results

In this section, we first introduce necessary preliminaries, including basic assumptions and definitions, then present D-OSPA and its theoretical guarantees.

Preliminaries

Following previous studies in centralized OCO (Hazan 2016) and D-OCO (Hosseini, Chapman, and Mesbahi 2013), we introduce three foundational assumptions.

Assumption 1. *The convex decision set $\mathcal{K} \subset \mathbb{R}^d$ is convex and compact. Furthermore, the norm of any points in the set is upper bounded by D, i.e.,*

$$\forall \mathbf{x} \in \mathcal{K}, \|\mathbf{x}\|_2 \leq D. \tag{7}$$

Assumption 2. *At each round t, the local loss function $f_{t,i}(\mathbf{x})$ is G-Lipschitz over $\mathcal{K}$ for any learner i, i.e.,*

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}, |f_{t,i}(\mathbf{x}) - f_{t,i}(\mathbf{y})| \leq G\|\mathbf{x} - \mathbf{y}\|_2. \tag{8}$$

Assumption 3. *In the distributed network, learners communicate with neighbors by a non-negative, symmetric and doubly stochastic matrix $P \in \mathbb{R}^{n \times n}$, which guarantees*

- $\forall\, 0 \leq i, j \leq n,\ 0 \leq P_{ij} \leq 1$;
- $\forall\, 0 \leq i, j \leq n,\ P_{ij} > 0$ *if and only if* $(i, j) \in E$;
- $\forall\, 0 \leq i \leq n,\ \sum_{j=1}^{n} P_{ij} = \sum_{j \in N_i} P_{ij} = 1$;
- $\forall\, 0 \leq j \leq n,\ \sum_{i=1}^{n} P_{ij} = \sum_{i \in N_j} P_{ij} = 1$.

In the following, we denote $\sigma_2(P)$ as the second largest eigenvalue of the matrix P.

In this work, we assume that the adversary is oblivious, i.e., all local loss functions are chosen in advance and are independent of the actions played by local learners.

Then, we recall standard definitions of convexity, smoothness (Boyd and Vandenberghe 2004) and linear optimization oracle (Hazan and Minasyan 2020).

Definition 1. *Let $f(\mathbf{x}) : \mathcal{K} \rightarrow \mathbb{R}$ be a function over $\mathcal{K}$. It is called convex over $\mathcal{K}$ if for all $\mathbf{x} \in \mathcal{K}, \mathbf{y} \in \mathcal{K}$*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}). \tag{9}$$

Definition 2. *Let $f(\mathbf{x}) : \mathcal{K} \rightarrow \mathbb{R}$ be a function over $\mathcal{K}$. It is called β-smooth over $\mathcal{K}$ if for all $\mathbf{x} \in \mathcal{K}, \mathbf{y} \in \mathcal{K}$*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\beta}{2}\|\mathbf{y} - \mathbf{x}\|_2^2. \tag{10}$$

Definition 3. *Let $\mathcal{O}_{\mathcal{K}}(\cdot)$ be a linear optimization oracle over $\mathcal{K}$ and its corresponding linear value oracle is denoted as $\mathcal{M}_{\mathcal{K}}(\cdot)$. Then, for any $\mathbf{y} \in \mathbb{R}^d$, we have*

$$\mathcal{O}_{\mathcal{K}}(\mathbf{y}) = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{K}} \langle \mathbf{y}, \mathbf{x} \rangle\ ,\ \mathcal{M}_{\mathcal{K}}(\mathbf{y}) = \max_{\mathbf{x} \in \mathcal{K}} \langle \mathbf{y}, \mathbf{x} \rangle\ . \tag{11}$$

Algorithm 1: <u>D</u>istributed <u>F</u>ollow-the-<u>P</u>erturbed-<u>L</u>eader (D-FPL)

Input: Number of rounds T, communication matrix P, local learner set V, perturbation parameter η, linear optimization oracle $\mathcal{O}_{\mathcal{K}}(\cdot)$
Initialization: Set $\{\mathbf{z}_{0,i} = \mathbf{0} | i \in V\}$

for $t = 1$ to T **do**
  for $i \in V$ **do**
    compute $\mathbf{x}_{t,i}$ according to (12)
    play $\mathbf{x}_{t,i}$, observe $f_{t,i}$, denote $\nabla_{t,i} = \nabla f_{t,i}(\mathbf{x}_{t,i})$
    update $\mathbf{z}_{t,i}$ according to (6)
  end for
end for

Remark 1. We solve a linear optimization by querying the linear optimization oracle $\mathcal{O}_{\mathcal{K}}(\cdot)$. Notice that $\mathcal{M}_{\mathcal{K}}(\mathbf{y}) = \langle \mathbf{y}, \mathcal{O}_{\mathcal{K}}(\mathbf{y}) \rangle$ and $\nabla \mathcal{M}_{\mathcal{K}}(\mathbf{y}) = \mathcal{O}_{\mathcal{K}}(\mathbf{y})$.

In the following, for simplicity, we denote $a_{l:k}$ as the sum $a_{l:k} = \sum_{i=l}^{k} a_i$ and $\nabla_{t,i}$ as the local gradient $\nabla_{t,i} = \nabla f_{t,i}(\mathbf{x}_{t,i})$ of learner i at round t.

Methods

Our method is inspired by Hazan and Minasyan (2020), who proposed a projection-free method based on Follow-the-Perturbed-Leader (FPL) with an improved regret for smooth and convex losses in centralized OCO. To exploit the smoothness of losses in the distributed setting, we first present a distributed extension of FPL. Then, we introduce sampling and blocking techniques to reduce the computational cost. Eventually, we obtain our final method, named Distributed Online Smooth Projection-Free Algorithm (D-OSPA).

Distributed Follow-the-Perturbed-Leader

The classical FPL method can not be used in D-OCO directly since local learners can only access local gradients but need to minimize global losses. Following previous studies in D-OCO (Hosseini, Chapman, and Mesbahi 2013; Zhang et al. 2017), we introduce a dual variable $\mathbf{z}_{t,i}$ for each learner i to approximate the historical accumulative global gradients. At round t, learner i chooses action $\mathbf{x}_{t,i}$ according to

$$\mathbf{x}_{t,i} = \mathbb{E}_{\mathbf{v} \sim \mathbb{B}} \left[\mathcal{O}_{\mathcal{K}} \left(-\mathbf{z}_{t-1,i} + \frac{1}{\eta} \cdot \mathbf{v} \right) \right] \tag{12}$$

and updates $\mathbf{z}_{t,i}$ by (6). By replacing (5) in FPL with (12) and maintaining the dual variable according to (6), we obtain the distributed extension of FPL, termed as Distributed Follow-the-Perturbed-Leader (D-FPL). The complete procedures are summarized in Algorithm 1, of which the regret bound is stated as following.

Theorem 1. *Let $L = \frac{1}{2} + \frac{3\sqrt{n}}{1-\sigma_2(P)}$. Under Assumptions 1, 2 and 3, if the loss functions are convex, for any $i \in V$, Algorithm 1 guarantees*

$$\text{Regret}_i \leq \frac{2Dn}{\eta} + \eta d D G^2 T n L. \tag{13}$$

Algorithm 2: Distributed Sampled Follow-the-Perturbed-Leader (D-SFPL)

Input: Number of rounds T, communication matrix P, local learner set V, perturbation parameter η, number of samples m, linear optimization oracle $\mathcal{O}_{\mathcal{K}}(\cdot)$
Initialization: Set $\{\tilde{\mathbf{z}}_{0,i} = \mathbf{0} | i \in V\}$

```
for t = 1 to T do
    for i ∈ V do
        compute x̃^u_{t,i} according to (14) for u = 1 to m
        play x̃_{t,i} = (1/m) Σ_{u=1}^m x̃^u_{t,i}, observe f_{t,i} and denote
            ∇̃_{t,i} = ∇f_{t,i}(x̃_{t,i})
        update z̃_{t,i} according to (15)
    end for
end for
```

Especially, when $\eta = \frac{1}{G}\sqrt{\frac{2}{dLT}}$, Algorithm 1 guarantees $\mathrm{Regret}_i \leq 2DGn\sqrt{2dLT} = \mathcal{O}(\sqrt{T})$.

Remark 2. Theorem 1 indicates that if losses are convex (without the smoothness condition), D-FPL enjoys the optimal regret of $\mathcal{O}(\sqrt{T})$ when $\eta = \mathcal{O}(T^{-1/2})$. To the best of our knowledge, D-FPL is the first distributed extension of FPL.

Distributed Sampled Follow-the-Perturbed-Leader

Notice that the expectation operation in D-FPL is computationally intractable in practice. To deal with this issue, we propose Distributed Sampled Follow-the-Perturbed-Leader (D-SFPL), which only solves m linear optimizations for each local learner per round via the sampling technique (Cohen and Hazan 2015; Hazan and Minasyan 2020).

More clearly, at round t each local learner i samples m random vectors $\mathbf{v}^u_{t,i}$ ($u = 1, \cdots m$) from the unit ball $\mathbb{B}$, computes m temporary decisions

$$\tilde{\mathbf{x}}^u_{t,i} = \mathcal{O}_{\mathcal{K}}\left(-\tilde{\mathbf{z}}_{t-1,i} + \frac{1}{\eta} \cdot \mathbf{v}^u_{t,i}\right) \tag{14}$$

and plays $\tilde{\mathbf{x}}_{t,i} = \frac{1}{m}\sum_{u=1}^m \tilde{\mathbf{x}}^u_{t,i}$ as the final decision. Then, learner i maintains $\tilde{\mathbf{z}}_{t,i}$ as following

$$\tilde{\mathbf{z}}_{t,i} = \sum_{j \in N_i} P_{ij}\tilde{\mathbf{z}}_{t-1,j} + \tilde{\nabla}_{t,i}, \tag{15}$$

where P is the communication matrix. Therefore, D-SFPL only solves $m \cdot T$ linear optimizations, which is more computationally efficient in comparison with D-FPL. We summarize the detailed procedure in Algorithm 2 and present expected regret bounds of D-SFPL in the following theorem.

Theorem 2. *Let $L = \frac{1}{2} + \frac{3\sqrt{n}}{1-\sigma_2(P)}$.Under Assumptions 1, 2 and 3, if the loss functions are convex, for any $i \in V$, Algorithm 2 guarantees an expected regret of*

$$\mathbb{E}\left[\mathrm{Regret}_i\right] \leq \frac{2Dn}{\eta} + \eta dDG^2TnL + \frac{6DGTn}{\sqrt{m}}. \tag{16}$$

Under the same assumptions, if the loss functions are convex and β-smooth , Algorithm 2 guarantees an expected regret of

$$\mathbb{E}\left[\mathrm{Regret}_i\right] \leq \frac{2Dn}{\eta} + \eta dDG^2TnL + \frac{8\beta D^2Tn}{m}. \tag{17}$$

Remark 3. Theorem 2 suggests that when $\eta = \mathcal{O}(T^{-1/2})$, D-SFPL guarantees an expected regret of $\mathcal{O}(\sqrt{T})$ with $m = T$ for general convex losses and $m = \sqrt{T}$ for smooth and convex losses.

Although Theorem 2 provides the expected regret bounds for D-SFPL, we still wonder whether it can enjoy the bounds most of the time. For this reason, we present high-probability regret bounds, which imply that D-SFPL guarantees almost the same bounds (up to logarithmic factors) as that in Theorem 2.

Theorem 3. *Let $L = \frac{1}{2} + \frac{3\sqrt{n}}{1-\sigma_2(P)}$, $r = 2D\sqrt{\frac{2}{m}\ln\frac{2T}{\delta}}$ and $r' = 2DG\sqrt{2T\ln\frac{4}{\delta}}$. Under Assumptions 1, 2 and 3, if the loss functions are convex, for any $i \in V$ and for any $\delta > 0$, Algorithm 2 with probability $1-\delta$ guarantees a regret of*

$$\mathrm{Regret}_i \leq \frac{2Dn}{\eta} + \eta dDG^2TnL + 3rGTn. \tag{18}$$

Under the same assumption, if the loss functions are convex and β-smooth , Algorithm 2 with probability $1-\delta$ guarantees a regret of

$$\mathrm{Regret}_i \leq \frac{2Dn}{\eta} + \eta dDG^2TnL + 3r'n + 2\beta r^2Tn. \tag{19}$$

Distributed Online Smooth Projection-Free Algorithm

In each round, D-SFPL needs to solve m linear optimizations where m may scale with T. This still leads to a high computational cost. In the following, we investigate how to decrease m to $\mathcal{O}(1)$ to promote the efficiency.

The key idea to handle this issue is the blocking technique (Garber and Kretzu 2020; Hazan and Minasyan 2020), which divides the whole rounds into size-equivalent blocks and only updates actions at the beginning of each block. By applying this technique to D-SFPL, we propose Distributed Online Smooth Projection-Free Algorithm (D-OSPA). It should be noticed that D-OSPA updates $\tilde{\mathbf{z}}_{t,i}$ with all local gradients in the same block instead of only one:

$$\tilde{\mathbf{z}}_{t,i} = \sum_{j \in N_i} P_{ij}\tilde{\mathbf{z}}_{t-1,j} + \tilde{\nabla}_{t-k:t-1,i}, \tag{20}$$

where $\tilde{\nabla}_{t-k:t-1,i}$ is the sum of all local gradients for leaner i in one block (from round $t-k$ to round $t-1$).

We can derive the regret bounds of D-OSPA from previous theorems by using reductions, e.g., aggregating local loss functions $f_{t,i}$ for learner i in the same block into a new loss function $f'_{t',i}$. The reduced game has $T' = T/k$ rounds (without loss of generality we assume T' is an integer), where k is the block size and T is the number of whole rounds in the original game. Here, we list some crucial changes in the reduced game as follows.

- In Assumption 1, the domain set in the reduced game is upper bounded by $D' = D$.
- In Assumption 2, for the block loss function $f'_{t',i} = \sum_{t=(t'-1)\cdot k+1}^{t'\cdot k} f_{t,i}$, the Lipschitz constant in the reduced game is $G' = k \cdot G$.

Algorithm 3: Distributed Online Smooth Projection-Free Algorithm (D-OSPA)

Input: Number of rounds T, communication matrix P, local learner set V, perturbation parameter η, number of sample m, block size k, linear optimization oracle $\mathcal{O}_{\mathcal{K}}(\cdot)$

Initialization: For any local learner i, set $\tilde{\mathbf{z}}_{0,i} = \mathbf{0}$ and choose arbitrary $\tilde{\mathbf{x}}_{0,i} \in \mathcal{K}$

```
for t = 1 to T do
  for i ∈ V do
    if t mod k = 1 and t > 1 then
      update z̃_{t,i} according to (20)
      compute x̃^u_{t,i} according to (14) for u = 1 to m
      play x̃_{t,i} = (1/m) Σ_{u=1}^{m} x̃^u_{t,i}, observe f_{t,i} and denote ∇̃_{t,i} = ∇f_{t,i}(x̃_{t,i})
    else
      update z̃_{t,i} = z̃_{t-1,i}
      play x̃_{t,i} = x̃_{t-1,i}, observe f_{t,i} and denote ∇̃_{t,i} = ∇f_{t,i}(x̃_{t,i})
    end if
  end for
end for
```

- If $f_{t,i}$ is β-smooth, then the block loss function $f'_{t',i} = \sum_{t=(t'-1)\cdot k+1}^{t'\cdot k} f_{t,i}$ is β'-smooth where $\beta' = k \cdot \beta$.

After above reductions, the reduced game still guarantees Theorem 2 and Theorem 3 with the domain bound D', the gradient bound G', the total rounds T', the local loss function $f'_{t,i}$ and the smoothness parameter β'. Then, we obtain the following theorems.

Theorem 4. *Let $L = \frac{1}{2} + \frac{3\sqrt{n}}{1-\sigma_2(P)}$, $m = k = T^{1/2}$ and $\eta = \frac{1}{kG}\sqrt{\frac{2}{dL}T^{-1/2}}$. Under Assumptions 1, 2 and 3, if the loss functions are convex, for any $i \in V$, Algorithm 3 guarantees an expected regret of*

$$\mathbb{E}\left[\text{Regret}_i\right] \le nGD\left(2\sqrt{2dL}+6\right)T^{\frac{3}{4}} = \mathcal{O}\left(T^{\frac{3}{4}}\right). \quad (21)$$

Let $m = k = T^{1/3}$ and $\eta = \frac{1}{kG}\sqrt{\frac{2}{dL}T^{-2/3}}$. Under the same assumption, if the loss functions are convex and β-smooth , Algorithm 3 guarantees an expected regret of

$$\mathbb{E}\left[\text{Regret}_i\right] \le nD\left(2G\sqrt{2dL}+8\beta D\right)T^{\frac{2}{3}} = \mathcal{O}\left(T^{\frac{2}{3}}\right). \quad (22)$$

Theorem 5. *Under Assumptions 1, 2 and 3, if the loss functions are convex , for any $i \in V$ and for any $\delta > 0$, Algorithm 3 with probability $1-\delta$ guarantees a regret of*

$$\text{Regret}_i = \tilde{\mathcal{O}}\left(T^{\frac{3}{4}}\ln\frac{1}{\delta}\right). \quad (23)$$

Under the same assumption, if the loss functions are convex and β-smooth , Algorithm 3 with probability $1-\delta$ guarantees a regret of

$$\text{Regret}_i = \tilde{\mathcal{O}}\left(T^{\frac{2}{3}}\ln\frac{1}{\delta}\right). \quad (24)$$

Remark 4. Above theorems imply that D-OSPA has almost the same regret as OSPF (Hazan and Minasyan 2020). However, D-OSPA is applied in distributed scenarios and therefore, is more scalable.

Remark 5. With the parameter choice of $m = k$, we only need to solve one linear optimization in each round on average. Hence, D-OSPA is more efficient than D-SFPL.

We notice that for smooth and convex losses, D-OSPA needs to communicate with neighbors $\mathcal{O}(T^{2/3})$ times due to the $T^{1/3}$ block size. While D-BOCG (Wan, Tu, and Zhang 2020) needs only $\mathcal{O}(T^{1/2})$ due to the $T^{1/2}$ block size. However, it does not imply that D-OSPA has a higher communication complexity because their regret bounds are different. To be more clear, we use the metric T_ϵ introduced by Hazan and Minasyan (2020): the number of computing gradients and solving linear optimizations during the whole rounds until the average regret is ϵ at most. In the case of smooth and convex losses, D-OSPA achieves $T_\epsilon = \mathcal{O}(d/\epsilon^3)$ with $\mathcal{O}(d/\epsilon^2)$ communication complexity, while D-BOCG achieves $T_\epsilon = \mathcal{O}(1/\epsilon^4)$ with $\mathcal{O}(1/\epsilon^2)$ communication complexity. Thus, under almost equal communication complexity (up to the dimension d), D-OSPA actually promotes the convergence rate by exploiting the smoothness of losses. The dimension dependence is a byproduct of Follow-the-Perturbed-Leader and whether it can be removed is still an open problem (Hazan and Minasyan 2020).

Theoretical Analysis

The extension of previous methods is intuitive, and the main challenge of this work lies in theoretical analysis. We summarize the major difficulty as following.

> Each learner cannot access local information of all other learners in the distributed network. Hence, it is difficult to bound the global regret on each leaner separately.

To cope with this problem, we establish the connections between any local learner i and a virtual centralized learner. Then, the global regret analysis can be converted to this virtual learner without loosing too much.

Due to the limitation of space, we only present the detailed proof of Theorem 1. The omitted proofs can be found in the supplementary material.

Proof of Theorem 1

In the beginning, we review the standard definition of Fenchel Dual (Boyd and Vandenberghe 2004).

Definition 4. *Let $f(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$ be a function over $\mathcal{K}$. The Fenchel dual of $f(\mathbf{x})$ is defined as $f^*(\mathbf{y})$, which satisfies*

$$\forall \mathbf{y} \in \mathbb{R}^d, f^*(\mathbf{y}) = \sup_{\mathbf{x}\in\mathcal{K}} \left\{\langle \mathbf{y}, \mathbf{x}\rangle - f(\mathbf{x})\right\}. \quad (25)$$

For brevity, we denote $h^*_\eta(\mathbf{y}) = \mathbb{E}_{\mathbf{v}\sim\mathbb{B}}\left[\mathcal{M}_{\mathcal{K}}(\mathbf{y} + \frac{1}{\eta}\cdot\mathbf{v})\right]$, which is described as the Fenchel Dual of the regularization $h_\eta(\mathbf{y})$ in the analysis framework. According to $\nabla\mathcal{M}_{\mathcal{K}}(\mathbf{y}) = \mathcal{O}_{\mathcal{K}}(\mathbf{y})$, we can obtain $\nabla h^*_\eta(\mathbf{y}) = \mathbb{E}_{\mathbf{v}\sim\mathbb{B}}\left[\mathcal{O}_{\mathcal{K}}(\mathbf{y} + \frac{1}{\eta}\cdot\mathbf{v})\right]$.

Then, we introduce $\bar{\mathbf{z}}_{t-1} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{z}_{t-1,j}$ as the average of all dual variables and $\bar{\mathbf{x}}_t = \nabla h_\eta^*(-\bar{\mathbf{z}}_{t-1})$ as the decision made by a virtual centralized learner. According to the following lemma, we can upper bound the distance between the decision of local learner i and that of the virtual one.

Lemma 1. *Let $\mathbf{z}_{t,i}$ and $\mathbf{x}_{t,i}$ be defined as that in Algorithm 1. Denote $\bar{\mathbf{z}}_{t-1} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{z}_{t-1,j}$ and $\bar{\mathbf{x}}_t = \nabla h_\eta^*(-\bar{\mathbf{z}}_{t-1})$, then we have*

$$\|\bar{\mathbf{x}}_t - \mathbf{x}_{t,i}\|_2 \le \epsilon, \tag{26}$$

where $\epsilon = \eta d D \frac{\sqrt{n}G}{1-\sigma_2(P)}$.

Applying Lemma 1, the global regret for Algorithm 1 can be upper bounded by a temporary term with respect to the virtual learner, as stated below.

Lemma 2. *Let $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}}\sum_{j=1}^{n}\sum_{t=1}^{T} f_{t,j}(\mathbf{x})$ and $\bar{\mathbf{x}}_t = \nabla h_\eta^*(-\bar{\mathbf{z}}_{t-1})$. Then, we have*

$$\begin{aligned}\mathrm{Regret}_i &= \sum_{j=1}^{n}\sum_{t=1}^{T}\left[f_{t,j}(\mathbf{x}_{t,i}) - f_{t,j}(\mathbf{x}^*)\right] \\ &\le n\sum_{t=1}^{T}\left\langle \bar{\nabla}_t, \bar{\mathbf{x}}_t - \mathbf{x}^*\right\rangle + 3\epsilon G T n,\end{aligned} \tag{27}$$

in which $\bar{\nabla}_t = \frac{1}{n}\sum_{j=1}^{n}\nabla_{t,j}$ and $\epsilon = \eta d D\frac{\sqrt{n}G}{1-\sigma_2(P)}$.

Remark 6. Lemma 2 indicates that the global regret of Algorithm 1 is upper bounded by a linear term. This term can be viewed as the difference between the loss with $\bar{\mathbf{x}}_t$ made by the virtual learner and the fixed decision $\mathbf{x}^*$, where the loss function is $F_t(\mathbf{x}) = \langle\bar{\nabla}_t, \mathbf{x}\rangle$. Therefore, we can convert the theoretical analysis to this virtual learner, without loosing too much in the global regret.

Next, to upper bound $\sum_{t=1}^{T}\langle\bar{\nabla}_t, \bar{\mathbf{x}}_t - \mathbf{x}^*\rangle$, we follow the primal-dual framework (Shalev-Shwartz and Singer 2007, 2006) and consider the primal optimization problem as described below

$$\min_{\mathbf{x}\in\mathcal{K}}\left\{h_\eta(\mathbf{x}) + \sum_{t=1}^{T}F_t(\mathbf{x})\right\}, \tag{28}$$

where $F_t(\mathbf{x})$ is defined as $F_t(\mathbf{x}) = \langle\bar{\nabla}_t, \mathbf{x}\rangle$ and $h_\eta(\mathbf{x})$ is viewed as the regularization. By using Lagrange multipliers, the dual problem of (28) is placed as follows

$$\max_{\bar{\lambda}_1,\cdots,\bar{\lambda}_T} D(\bar{\lambda}_1,\cdots,\bar{\lambda}_T) = -h_\eta^*\left(-\sum_{t=1}^{T}\bar{\lambda}_t\right) - \sum_{t=1}^{T}F_t^*(\bar{\lambda}_t). \tag{29}$$

Details can be found in Shalev-Shwartz and Singer (2006). By weak duality, the dual problem (29) is upper bounded by (28). Hence, it is natural to get the solution of (28) by increasing $D(\bar{\lambda}_1,\cdots,\bar{\lambda}_T)$ with different $\bar{\lambda}_r (r = 1,\cdots,T)$. Following Shalev-Shwartz and Singer (2006), we choose $D(\bar{\lambda}_1^t,\cdots,\bar{\lambda}_r^t,\bar{\lambda}_{r+1}^t,\cdots,\bar{\lambda}_T^t) = D(\bar{\nabla}_1,\cdots,\bar{\nabla}_t,0,\cdots,0)$ at round t, which means for $D(\bar{\lambda}_1^t,\cdots,\bar{\lambda}_T^t)$ we have

$$\bar{\lambda}_r^t = \begin{cases}\bar{\nabla}_r, & \text{if } r \le t \\ 0, & \text{if } r > t\end{cases}. \tag{30}$$

Furthermore, according to the definition of dual problem (29), we can obtain

$$\begin{aligned}&D(\bar{\lambda}_1^t,\cdots,\bar{\lambda}_T^t) = D(\bar{\nabla}_1,\cdots,\bar{\nabla}_t,0,\cdots,0) \\ &= -h_\eta^*\left(-\bar{\nabla}_{1:t}\right) - \sum_{r=1}^{t}F_r^*(\bar{\nabla}_r) - \sum_{r=t+1}^{T}F_r^*(0).\end{aligned} \tag{31}$$

For convenience, denote the optimal solution of (29) as $\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^* = \operatorname{argmax}_{\bar{\lambda}_1,\cdots,\bar{\lambda}_T} D(\bar{\lambda}_1,\cdots,\bar{\lambda}_T)$. And we can upper bound of $D(\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^*)$, as described below.

Lemma 3. *Let $\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^*$ be the optimal solution of $D(\bar{\lambda}_1,\cdots,\bar{\lambda}_T)$. Then, we have*

$$D(\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^*) \le \frac{D}{\eta} + \min_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}F_t(\mathbf{x}). \tag{32}$$

Now, consider $D(\bar{\lambda}_1^T,\cdots,\bar{\lambda}_T^T) = D(\bar{\nabla}_1,\cdots,\bar{\nabla}_T)$, the value of the dual problem (29) at round T. We can provide a lower bound of $D(\bar{\nabla}_1,\cdots,\bar{\nabla}_T)$, as stated below.

Lemma 4. *Under Assumptions 1 and 2, we have*

$$D(\bar{\nabla}_1,\cdots,\bar{\nabla}_T) \ge \sum_{t=1}^{T}F_t(\bar{\mathbf{x}}_t) - \frac{\eta d D}{2}G^2 T - \frac{D}{\eta}. \tag{33}$$

In fact, according to the definition of $\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^*$ and Lemma 3, we have

$$D(\bar{\nabla}_1,\cdots,\bar{\nabla}_T) \le D(\bar{\lambda}_1^*,\cdots,\bar{\lambda}_T^*) \overset{(32)}{\le} \frac{D}{\eta} + \min_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}F_t(\mathbf{x}). \tag{34}$$

Substituting (33) into (34), we have

$$\begin{aligned}\sum_{t=1}^{T}F_t(\bar{\mathbf{x}}_t) - \sum_{t=1}^{T}F_t(\mathbf{x}^*) &\le \sum_{t=1}^{T}F_t(\bar{\mathbf{x}}_t) - \min_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}F_t(\mathbf{x}) \\ &\le \frac{\eta d D}{2}G^2 T + \frac{2D}{\eta},\end{aligned} \tag{35}$$

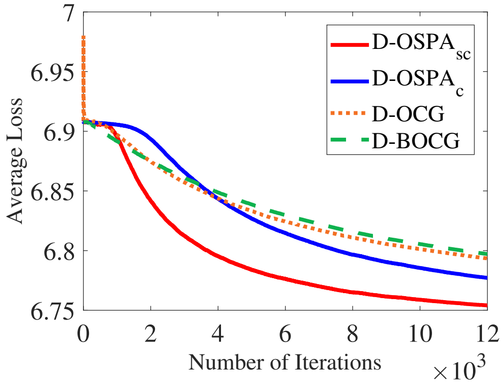
where $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}}\sum_{j=1}^{n}\sum_{t=1}^{T}f_{t,j}(\mathbf{x})$.

Applying Lemma 2 and (35) , we have

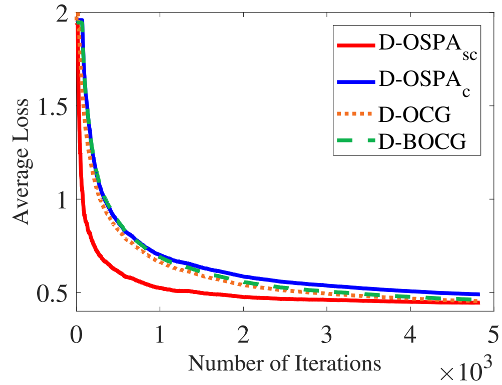
$$\begin{aligned}\mathrm{Regret}_i &\le n\sum_{t=1}^{T}\left\langle\bar{\nabla}_t, \bar{\mathbf{x}}_t - \mathbf{x}^*\right\rangle + 3\epsilon G T n \\ &= n\sum_{t=1}^{T}\left[F_t(\bar{\mathbf{x}}_t) - F_t(\mathbf{x}^*)\right] + 3\epsilon G T n \\ &\overset{(35)}{\le} n\left\{\frac{\eta d D}{2}G^2 T + \frac{2D}{\eta}\right\} + 3\epsilon G T n \\ &= \frac{2Dn}{\eta} + \eta d D G^2 T n L,\end{aligned} \tag{36}$$

where $L = \frac{1}{2} + \frac{3\sqrt{n}}{1-\sigma_2(P)}$. If $\eta = \frac{1}{G}\sqrt{\frac{2}{dLT}}$, we obtain that

$$\mathrm{Regret}_i = 2DGn\sqrt{2dLT} = \mathcal{O}(\sqrt{T}). \tag{37}$$



(a) aloi



(b) shuttle

Figure 1: Experiments on benchmark datasets.

Experiments

In this section, we provide experimental results on benchmark datasets to illustrate the empirical performance of our proposed method.

Experimental Settings

Following Zhang et al. (2017), we conduct the simulation experiments for multiclass classification in D-OCO. At round t, local learner i receives a training example $e_{t,i} \in \mathbb{R}^k$ from class $y_{t,i} \in \{1, \cdots, h\}$. Then, learner i chooses a decision $\mathbf{X}_{t,i} = [\mathbf{x}_1^\top; \cdots; \mathbf{x}_h^\top]$ from $\mathcal{K} = \{\mathbf{X} \in \mathbb{R}^{h\times k} : \|\mathbf{X}\|_* \leq \tau\}$, where $\|\cdot\|_*$ is the trace norm of matrices and τ is a constant. After that, learner i predicts the class label of $e_{t,i}$ with $\text{argmax}_{l\in[h]} \mathbf{x}_l^\top e_{t,i}$ and incurs a multivariate logistic loss

$$f_{t,i}(\mathbf{X}_{t,i}) = \log\left(1 + \sum_{l\neq y_{t,i}} \exp\left(\mathbf{x}_l^\top e_{t,i} - \mathbf{x}_{y_{t,i}}^\top e_{t,i}\right)\right),$$

which is a smooth and convex function. In our experiments, learners are connected via a cycle graph with 9 nodes, which indicates that local learner i has only two neighbors. Correspondingly, the communication matrix P is set as $P_{ij} = 1/3$ if $j \in N(i)$ or $j = i$, which satisfies Assumption 3. Besides, we use the average loss $\frac{1}{tn^2}\sum_{r=1}^{t}\sum_{i=1}^{n}\sum_{j=1}^{n} f_{r,j}(\mathbf{X}_{r,i})$ introduced by Wan, Tu, and Zhang (2020) to measure the performance of each method at round t.

Experimental Results

We choose D-OCG (Zhang et al. 2017) and D-BOCG (Wan, Tu, and Zhang 2020) as baseline methods. In details, we set $\tau = 10$ and the parameters of each method are set according to their theoretical suggestions. For D-OCG, $\sigma_{t,i} = 1/\sqrt{t}$ and $\eta = cT^{-3/4}$. For D-BOCG, $K = \lfloor T^{1/2} \rfloor$, $L_\epsilon = 20$ and $\eta = cT^{-3/4}$. For D-OSPA, we conduct two versions:

- D-OSPA$_{\text{sc}}$ for smooth and convex losses with $m = k = \lfloor T^{1/3} \rfloor$ and $\eta = cT^{-2/3}$;
- D-OSPA$_{\text{c}}$ for general convex losses with $m = k = \lfloor T^{1/2} \rfloor$ and $\eta = cT^{-3/4}$.

Since D-OSPA is a randomized algorithm, we repeat the experiments 10 times and the average results are reported. The hyper-parameter c is selected from $\{2^{-3}, 2^{-2}, \cdots, 2^6\}$. The experiments are conducted on aloi and shuttle from LIBSVM repository (Chang and Lin 2011).

Fig. 1 shows the change of average loss against the number of iterations for each method. As can be seen, by exploiting the smoothness of loss functions, D-OSPA$_{\text{sc}}$ converges faster than baseline methods and achieves the lowest average loss on two datasets. When only utilizing the convexity of losses, D-OSPA$_{\text{c}}$ performs similarly with D-OCG and D-BOCG, which is reasonable because of their same theoretical guarantees.

Conclusion and Future Work

In this paper, we propose the first distributed variant of Follow-the-Perturbed-Leader, which achieves the optimal regret of $\mathcal{O}(\sqrt{T})$ in D-OCO. Then, we reduce the computational cost by utilizing sampling and blocking techniques. In this way, we obtain a new distributed projection-free method, termed as Distributed Online Smooth Projection-Free Algorithm (D-OSPA), with one linear optimization per round on average for each local learner. By exploiting the smoothness of loss functions, our D-OSPA achieves an $\mathcal{O}(T^{2/3})$ regret bound, which is better than previous projection-free methods in D-OCO. Finally, we conduct simulation experiments on benchmark datasets, which demonstrate the effectiveness of D-OSPA.

Currently, we only consider static regret. In the future, we will investigate dynamic regret for distributed projection-free online learning. We note that in the centralized setting, recent work (Wan, Xue, and Zhang 2021; Wan, Zhang, and Song 2023) have established dynamic bounds for projection-free methods. But in the distributed setting, it seems non-trivial, and we leave it as future work.

Acknowledgments

This work was partially supported by NSFC (62122037, 61921006), and the major key project of PCL (PCL2021A12). We would like to thank the anonymous reviewers for their constructive suggestions.

References

Bao, Y.; Peng, Y.; Wu, C.; and Li, Z. 2018. Online Job Scheduling in Distributed Machine Learning Clusters. In *4th IEEE Conference on Computer Communications*, 495–503.

Boyd, S.; and Vandenberghe, L. 2004. Convex Optimization. *Cambridge University Press*.

Bubeck, S. 2015. Convex Optimization: Algorithms and Complexity. *Foundations and Trends in Machine Learning*, 8(3-4): 231–357.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27): 1–27.

Cohen, A.; and Hazan, T. 2015. Following the Perturbed Leader for Online Structured Learning. In *Proceedings of the 32nd International Conference on Machine Learning*, 1034–1042.

Frank, M.; and Wolfe, P. 1956. An Algorithm for Quadratic Programming. *Naval research logistics quarterly*, 3(1-2): 95–110.

Garber, D.; and Hazan, E. 2016. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization. *SIAM Journal on Optimization*, 26(3): 1493–1528.

Garber, D.; and Kretzu, B. 2020. Improved Regret Bounds for Projection-free Bandit Convex Optimization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2196–2206.

Garber, D.; and Kretzu, B. 2022. New Projection-free Algorithms for Online Convex Optimization with Adaptive Regret Guarantees. In *Proceedings of the 35th Conference on Learning Theory*, 2326–2359.

Hazan, E. 2016. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4): 157–325.

Hazan, E.; Agarwal, A.; and Kale, S. 2007. Logarithmic Regret Algorithms for Online Convex Optimization. *Foundations and Trends in Machine Learning*, 69(2): 169–192.

Hazan, E.; and Kale, S. 2012. Projection-free Online Learning. In *Proceedings of the 29th International Conference on Machine Learning*, 1843–1850.

Hazan, E.; and Minasyan, E. 2020. Faster Projection-free Online Learning. In *Proceedings of the 33rd Conference on Learning Theory*, 1877–1893.

Hosseini, S.; Chapman, A.; and Mesbahi, M. 2013. Online Distributed Optimization via Dual Averaging. In *52nd IEEE Conference on Decision and Control*, 1484–1489.

Kalai, A.; and Vempala, S. 2005. Efficient Algorithms for Online Decision Problems. *Journal of Computer and System Sciences*, 71(3): 291–307.

Kretzu, B.; and Garber, D. 2021. Revisiting Projection-free Online Learning: the Strongly Convex Case. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 3592–3600.

Lee, S.; Nedić, A.; and Raginsky, M. 2016. Coordinate Dual Averaging for Decentralized Online Optimization with Nonseparable Global Objectives. *ArXiv e-prints*, arXiv:1508.07933v2.

Lesage-Landry, A.; and Callaway, D. S. 2020. Dynamic and Distributed Online Convex Optimization for Demand Response of Commercial Buildings. *IEEE Control Systems Letters*, 4(3): 632–637.

Levy, K.; and Krause, A. 2019. Projection Free Online Learning over Smooth Sets. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 1458–1466.

Li, D.; Wong, K. D.; Hu, Y. H.; and Sayeed, A. M. 2002. Detection, Classification and Tracking of Targets in Distributed Sensor Networks. *IEEE Signal Processing Magazine*, 19(2): 17–29.

Mhammedi, Z. 2022. Efficient Projection-Free Online Convex Optimization with Membership Oracle. In *Proceedings of the 35th Conference on Learning Theory*, 5314–5390.

Molinaro, M. 2020. Curvature of Feasible Sets in Offline and Online Optimization. *ArXiv e-prints*, arXiv:2002.03213.

Nesterov, Y. 2009. Primal-dual Subgradient Methods for Convex Problems. *Mathematical Programming*, 120(1): 261–283.

Orabona, F.; Cesa-Bianchi, N.; and Gentile, C. 2012. Beyond Logarithmic Bounds in Online Learning. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 823–831.

Ram, S. S.; Nedich, A.; and Veeravalli, V. V. 2010. Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization. *Journal of Optimization Theory and Applications*, 147(3): 516–545.

Shahrampour, S.; and Jadbabaie, A. 2018. Distributed Online Optimization in Dynamic Environments Using Mirror Descent. *IEEE Transactions on Automatic Control*, 63(3): 714–725.

Shalev-Shwartz, S.; and Singer, Y. 2006. Convex Repeated Games and Fenchel Duality. In *Advances in Neural Information Processing Systems 19*, 1265–1272.

Shalev-Shwartz, S.; and Singer, Y. 2007. A Primal-dual Perspective of Online Learning Algorithm. *Machine Learning*, 69(2-3): 115–142.

Srebro, N.; Sridharan, K.; and Tewari, A. 2010. Smoothness, Low Noise and Fast Rates. In *Advances in Neural Information Processing Systems 23*, 2199–2207.

Wan, Y.; Tu, W.-W.; and Zhang, L. 2020. Projection-free Distributed Online Convex Optimization with $\mathcal{O}(\sqrt{T})$ Communication Complexity. In *Proceedings of the 37th International Conference on Machine Learning*, 9818–9828.

Wan, Y.; Wang, G.; Tu, W.-W.; and Zhang, L. 2022. Projection-free Distributed Online Learning with Sublinear

Communication Complexity. *Journal of Machine Learning Research*, 23(172): 1–53.

Wan, Y.; Xue, B.; and Zhang, L. 2021. Projection-free Online Learning in Dynamic Environments. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence Advances*, 10067–10075.

Wan, Y.; and Zhang, L. 2021. Projection-free Online Learning over Strongly Convex Set. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence Advances*, 10076–10084.

Wan, Y.; Zhang, L.; and Song, M. 2023. Improved Dynamic Regret for Online Frank-Wolfe. *ArXiv e-prints*, arXiv:2302.05620.

Yan, F.; Sundaram, S.; Vishwanathan, S.; and Qi, Y. 2013. Distributed Autonomous Online Learning: Regrets and Intrinsic Privacy-Preserving Properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11): 2483–2493.

Yuan, K.; Ling, Q.; and Yin, W. 2016. On the Convergence of Decentralized Gradient Descent. *SIAM Journal on Optimization*, 26(3): 1835–1854.

Zhang, W.; Zhao, P.; Zhu, W.; Hoi, S. C. H.; and Zhang, T. 2017. Projection-free Distributed Online Learning in Networks. In *Proceedings of the 34th International Conference on Machine Learning*, 4054–4062.

Zinkevich, M. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 928–936.