Projection-free Distributed Online Convex Optimization with $O(\sqrt{T})$ Communication Complexity

Yuanyu Wan [1] **Wei-Wei Tu** [2] **Lijun Zhang** [1]

Abstract

To deal with complicated constraints via locally light computations in distributed online learning, a recent study has presented a projection-free algorithm called distributed online conditional gradient (D-OCG), and achieved an $O(T^{3/4})$ regret bound, where T is the number of prediction rounds. However, in each round, the local learners of D-OCG need to communicate with their neighbors to share the local gradients, which results in a high communication complexity of $O(T)$. In this paper, we first propose an improved variant of D-OCG, namely D-BOCG, which enjoys an $O(T^{3/4})$ regret bound with only $O(\sqrt{T})$ communication complexity. The key idea is to divide the total prediction rounds into $\sqrt{T}$ equally-sized blocks, and only update the local learners at the beginning of each block by performing iterative linear optimization steps. Furthermore, to handle the more challenging bandit setting, in which only the loss value is available, we incorporate the classical one-point gradient estimator into D-BOCG, and obtain similar theoretical guarantees.

1. Introduction

Conditional gradient (CG) (Frank & Wolfe, 1956) is a simple yet efficient offline algorithm for solving high-dimensional problems with complicated constraints. To find a feasible solution, instead of performing the time-consuming projection step, CG utilizes the linear optimization step, which can be carried out much more efficiently. For example, in the matrix completion problem (Hazan & Kale, 2012), where the feasible set consists of all matrices with bounded trace norm, the projection step needs to compute the singular value decomposition (SVD) of a matrix. In contrast, the linear optimization step in CG only requires computing the top singular vector pair of a matrix, which is at least an order of magnitude faster than the SVD. Due to the emergence of large-scale problems, online conditional gradient (OCG) (Hazan & Kale, 2012; Hazan, 2016) was proposed for online convex optimization (OCO), which is viewed as a multi-round game between a learner and an adversary (Zinkevich, 2003). Different from CG that requires all data related to the objective function are given beforehand, this online variant can efficiently update the learner based on only a single data point in each round.

Recently, Zhang et al. (2017) further proposed D-OCG by extending OCG into a more practical scenario—distributed setting. It is well motivated by many distributed applications such as multi-agent coordination and distributed tracking in sensor networks (Li et al., 2002; Xiao et al., 2007; Nedić et al., 2009; Duchi et al., 2011). Specifically, the distributed setting is formulated as a set of local learners connected by an undirected graph, and each local learner can only communicate with its neighbors. The key idea of D-OCG is to maintain OCG for each local learner, and update it according to the local gradient as well as that received from its neighbors in each round. Compared with projection-based distributed algorithms (Ram et al., 2010; Hosseini et al., 2013), D-OCG significantly reduces the time cost for solving high-dimension problems with complicated constraints. Moreover, D-OCG is more scalable than OCG, since it can utilize many locally light computation resources to handle large-scale problems. However, because the local learners of D-OCG communicate with their neighbors to share the local gradients in each round, it suffers a high communication complexity of $O(T)$, where T is the total number of rounds.

In this paper, we first propose distributed block online conditional gradient (D-BOCG), an improved variant of D-OCG, which reduces the communication complexity from $O(T)$ to $O(\sqrt{T})$. To this end, we borrow the delayed update mechanism and the iterative linear optimization steps which have been employed to improve projection-free bandit convex optimization (Garber & Kretzu, 2019), and apply them to the distributed setting considered here. Specifically, according to the delayed update mechanism, we divide the total

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2]4Paradigm Inc., Beijing, China. Correspondence to: Lijun Zhang <zhanglj@lamda.nju.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

T rounds into $\sqrt{T}$ equally-sized blocks and only update the local learners at the beginning of each block. In this way, the local learners only need to communicate with their neighbors once for each block, which immediately implies the total communication complexity is $O(\sqrt{T})$. However, since the number of updates is decreased, only performing 1 linear optimization step as D-OCG for each update will increase the regret. To keep the same regret bound as that of D-OCG with less number of updates, we perform iterative linear optimization steps for each update. Theoretical analysis demonstrates that our D-BOCG achieves an $O(T^{3/4})$ regret bound with the number of linear optimization on the same order as that required by D-OCG.

Furthermore, to handle the more challenging bandit setting, we propose distributed block bandit conditional gradient (D-BBCG) by combining D-BOCG with the classical one-point gradient estimator (Flaxman et al., 2005), which can approximate the gradient with a single loss value. Similar to D-BOCG, the communication complexity of our D-BBCG is still $O(\sqrt{T})$. Our theoretical analysis reveals that D-BBCG enjoys a high-probability regret bound of $\widetilde{O}(T^{3/4})$[1] with the number of linear optimization on the same order as that required by D-OCG and D-BOCG.

2. Related Work

In this section, we briefly review the existing projection-free algorithms for OCO and its distributed variant.

2.1. Projection-free Algorithms for OCO

OCO is a general framework for online learning, which covers a variety of problems such as online portfolio selection (Blum & Kalai, 1999; Agarwal et al., 2006), online routing (Awerbuch & Kleinberg, 2004; 2008), online metric learning (Jain et al., 2008; Tsagkatakis & Savakis, 2011) and learning with expert advice (Cesa-Bianchi et al., 1997; Freund et al., 1997). It is generally viewed as a repeated game between a leaner and an adversary. In each round t, the learner first chooses a decision $\mathbf{x}(t)$ from a convex decision set $\mathcal{K} \subset \mathbb{R}^d$. Then, the adversary reveals a convex function $f_t(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$, which incurs a loss $f_t(\mathbf{x}(t))$ to the learner. The goal of the learner is to minimize the regret with respect to any fixed optimal decision, which is defined as

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{x}(t)) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

OCG (Hazan & Kale, 2012; Hazan, 2016) is the first projection-free algorithm for OCO, which enjoys the regret bound of $O(T^{3/4})$ and updates as the following linear optimization step

$$\begin{aligned} \mathbf{v} &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \left\{\nabla F_t(\mathbf{x}(t))^\top \mathbf{x}\right\} \\ \mathbf{x}(t+1) &= \mathbf{x}(t) + s_t(\mathbf{v} - \mathbf{x}(t)) \end{aligned} \tag{1}$$

where $F_t(\mathbf{x}) = \eta \sum_{k=1}^{t-1} \nabla f_k(\mathbf{x}(k))^\top \mathbf{x} + \|\mathbf{x} - \mathbf{x}(1)\|_2^2$, s_t and η are two parameters. Recently, projection-free algorithms with $O(\sqrt{T})$ regret were proposed for special decision sets such as polytope (Garber & Hazan, 2016) and smooth set (Levy & Krause, 2019). For smooth loss functions, a concurrent work (Hazan & Minasyan, 2020) proposed a randomized projection-free algorithm which achieves an expected regret bound of $O(T^{2/3})$.

Furthermore, OCG has been extended to handle the more challenging bandit setting, where only the loss value is available to the learner. Due to the lack of the gradient, Chen et al. (2019) proposed to combine OCG with the one-point gradient estimator (Flaxman et al., 2005) which can approximate the gradient with a single loss value, and established an expected regret bound of $O(T^{4/5})$ which is worse than the $O(T^{3/4})$ regret bound of OCG. Later, Garber & Kretzu (2019) proposed to divide the total rounds into several equally-sized blocks and perform iterative linear optimization steps at the beginning of each block, which improves the expected regret bound from $O(T^{4/5})$ to $O(T^{3/4})$. Additionally, we note that Chen et al. (2018) developed a projection-free algorithm for another interesting setting where the learner is allowed to access to the stochastic gradients.

2.2. Projection-free Algorithms for Distributed OCO

According to previous studies (Hosseini et al., 2013; Zhang et al., 2017), distributed OCO is a variant of OCO over a network defined by an undirected graph $G = (V, E)$, where $V = [n]$ is the node set and $E \subset V \times V$ is the edge set. Different from OCO where only exists 1 learner, in the distributed OCO, each node $i \in V$ is a local learner, and can only communicate with its immediate neighbors

$$N_i = \{j \in V | (i, j) \in E\}.$$

In each round t, each local learner $i \in V$ chooses a decision $\mathbf{x}_i(t) \in \mathcal{K}$, and then it receives a convex loss function $f_{t,i}(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$ chosen by the adversary. Moreover, the global loss function $f_t(\mathbf{x})$ is defined as the sum of local loss functions

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

The goal of each local learner $i \in V$ is to minimize the regret measured by the global loss with respect to the optimal fixed decision, which is defined as

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

[1] We use the $\widetilde{O}$ notation to hide constant factors as well as polylogarithmic factors in T.

Since the local loss function $f_{t,i}(\mathbf{x})$ is only available to the local learner i, to achieve this global goal for all local learners, it is necessary to utilize both their local gradients and those received from their neighbors.

Therefore, to make OCG distributed, Zhang et al. (2017) first introduce a non-negative weight matrix $P \in \mathbb{R}^{n\times n}$ and redefine $F_t(\mathbf{x})$ in OCG as

$$F_{t,i}(\mathbf{x}) = \eta \mathbf{z}_i(t)^\top \mathbf{x} + \|\mathbf{x} - \mathbf{x}_1(1)\|_2^2 \tag{2}$$

for each local learner i by replacing $\sum_{k=1}^{t-1} \nabla f_k(\mathbf{x}(k))$ with $\mathbf{z}_i(t)$, where $\mathbf{z}_i(1) = \mathbf{0}$ and

$$\mathbf{z}_i(t+1) = \sum_{j\in N_i} P_{ij}\mathbf{z}_j(t) + \nabla f_{t,i}(\mathbf{x}_i(t)). \tag{3}$$

Note that $\mathbf{z}_i(t)$ is a weighted sum of historical local gradients and those received from the neighbors, which could be viewed as an approximation for the sum of global gradients and is critical for minimizing the global regret.

Then, they proposed D-OCG updating as follows

$$\begin{aligned} &\textbf{for } \text{each local learner } i \in V \textbf{ do} \\ &\quad \mathbf{v}_i = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \left\{\nabla F_{t,i}(\mathbf{x}_i(t))^\top \mathbf{x}\right\} \\ &\quad \mathbf{x}_i(t+1) = \mathbf{x}_i(t) + s_t(\mathbf{v}_i - \mathbf{x}_i(t)) \\ &\textbf{end for} \end{aligned} \tag{4}$$

which enjoys $R_{T,i} = O(T^{3/4})$. However, in each round t, each local learner i needs to compute $\mathbf{z}_i(t+1)$ by communicating with its neighbors, which results in a high communication complexity of $O(T)$.

3. Main Results

In this section, we first introduce necessary preliminaries including common assumptions, definitions and a basic algorithmic ingredient. Then, we present our projection-free distributed online algorithm called D-BOCG for the full information setting, as well as its theoretical guarantee. Finally, we extend D-BOCG to the bandit setting.

3.1. Preliminaries

Following previous studies on OCO (Hazan & Kale, 2012; Garber & Kretzu, 2019) and the distributed OCO (Zhang et al., 2017), we first introduce the following assumptions.

Assumption 1 *At each round t, each local loss function $f_{t,i}(\mathbf{x})$ is G-Lipschitz over $\mathcal{K}$, i.e.,*

$$|f_{t,i}(\mathbf{x}) - f_{t,i}(\mathbf{y})| \le G\|\mathbf{x} - \mathbf{y}\|_2$$

for any $\mathbf{x} \in \mathcal{K}, \mathbf{y} \in \mathcal{K}$.

Assumption 2 *At each round t, each local loss function $f_{t,i}(\mathbf{x})$ is bounded over $\mathcal{K}$, i.e.,*

$$|f_{t,i}(\mathbf{x})| \le M$$

for any $\mathbf{x} \in \mathcal{K}$.

Assumption 3 *The convex decision set $\mathcal{K}$ is full dimensional and contains the origin. Moreover, there exist two constants $r, R > 0$ such that*

$$r\mathcal{B}^d \subseteq \mathcal{K} \subseteq R\mathcal{B}^d$$

where $\mathcal{B}^d$ denotes the unit Euclidean ball centered at the origin in $\mathbb{R}^d$.

Assumption 4 *The non-negative weight matrix $P \in \mathbb{R}^{n\times n}$ is symmetric and doubly stochastic, which satisfies*

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

The non-negative weight matrix P in Assumption 4 will be utilized to model the communication between the local learners in the distributed OCO, and we will use $\sigma_2(P)$ to denote the second largest eigenvalue of P.

Moreover, we recall the standard definitions for smooth and strongly convex functions (Boyd & Vandenberghe, 2004).

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

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

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

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2.$$

Let $f(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$ be an α-strongly convex function over $\mathcal{K}$ and $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x})$. Combining Definition 2 with the first order optimality condition (Boyd & Vandenberghe, 2004), it is easy to verify that

$$\frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}^*\|_2^2 \le f(\mathbf{x}) - f(\mathbf{x}^*) \tag{5}$$

for any $\mathbf{x} \in \mathcal{K}$, which was commonly used in previous studies (Hazan & Kale, 2012; Garber & Kretzu, 2019).

Finally, we present conditional gradient with stopping condition (CGSC) (Garber & Kretzu, 2019), which will be utilized as a subroutine of our proposed algorithms. Given a

Algorithm 1 CGSC

Input: feasible set $\mathcal{K}$, $\epsilon > 0$, L, $F(\mathbf{x})$, $\mathbf{x}_{\text{in}}$
$\tau = 0, \mathbf{c}_1 = \mathbf{x}_{\text{in}}$
repeat
  $\tau = \tau + 1$
  $\mathbf{v}_\tau \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \left\{\nabla F(\mathbf{c}_\tau)^\top \mathbf{x}\right\}$
  $s_\tau = \operatorname{argmin}_{s\in[0,1]} \{F(\mathbf{c}_\tau + s(\mathbf{v}_\tau - \mathbf{c}_\tau))\}$
  $\mathbf{c}_{\tau+1} = \mathbf{c}_\tau + s_\tau(\mathbf{v}_\tau - \mathbf{c}_\tau)$
until $\nabla F(\mathbf{c}_\tau)^\top(\mathbf{c}_\tau - \mathbf{v}_\tau) \leq \epsilon$ or $\tau = L$
return $\mathbf{x}_{\text{out}} = \mathbf{c}_\tau$

function $F(\mathbf{x}) : \mathcal{K} \to \mathbb{R}$ and an initial point $\mathbf{c}_1 = \mathbf{x}_{\text{in}} \in \mathcal{K}$, it iteratively performs the linear optimization step as follows

$$\mathbf{v}_\tau \in \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \left\{\nabla F(\mathbf{c}_\tau)^\top \mathbf{x}\right\}$$
$$\mathbf{c}_{\tau+1} = \mathbf{c}_\tau + s_\tau(\mathbf{v}_\tau - \mathbf{c}_\tau)$$

until $\nabla F(\mathbf{c}_\tau)^\top(\mathbf{c}_\tau - \mathbf{v}_\tau) \leq \epsilon$ or $\tau = L$, where ϵ, L are two parameters and

$$s_\tau = \operatorname*{argmin}_{s\in[0,1]} \{F(\mathbf{c}_\tau + s(\mathbf{v}_\tau - \mathbf{c}_\tau))\}$$

is selected by line search. The detailed procedures of CGSC are summarized in Algorithm 1.

Different from only performing linear optimization once, CGSC can output a point $\mathbf{x}_{\text{out}}$ such that $F(\mathbf{x}_{\text{out}})$ is very small when L is large enough and ϵ is small enough. As a result, it allows us to achieve a better regret bound than performing linear optimization once. Note that CGSC has been employed by Garber & Kretzu (2019) to develop a projection-free algorithm in the bandit setting, which attains the expected regret of $O(T^{3/4})$. In this paper, we introduce it into the distributed OCO to propose projection-free algorithms with only $O(\sqrt{T})$ communication complexity, and establish the $O(T^{3/4})$ and $\widetilde{O}(T^{3/4})$ regret for the full information and bandit settings, respectively.

3.2. Algorithm for Full Information Setting

To reduce the communication complexity of D-OCG, we first divide the total T rounds into B blocks of size K, where we assume that $B = T/K$ is an integer without loss of generality. Moreover, for each local learner $i \in V$, its decision in each block m stays the same and is denoted by $\mathbf{x}_i(m)$. In this way, the local gradient of local learner i in each round t is denoted by

$$\mathbf{g}_i(t) = \nabla f_{t,i}(\mathbf{x}_i(m_t))$$

where $m_t = \lceil t/K \rceil$, and the cumulative gradient of local learner i in each block m is denoted by

$$\widehat{\mathbf{g}}_i(m) = \sum_{t\in\mathcal{T}_m} \mathbf{g}_i(t)$$

Algorithm 2 D-BOCG

Input: feasible set $\mathcal{K}$, η, L, ϵ and K
Initialization: choose $\{\mathbf{x}_i(1) = \mathbf{0} \in \mathcal{K} | i \in V\}$ and set $\{\mathbf{z}_i(1) = \mathbf{0} | i \in V\}$
for $t = 1, \cdots, T$ **do**
  $m_t = \lceil t/K \rceil$
  for each local learner $i \in V$ **do**
    if $t > 1$ and $\text{mod}(t, K) = 1$ **then**
      $\widehat{\mathbf{g}}_i(m_t - 1) = \sum_{k=t-K}^{t-1} \mathbf{g}_i(k)$
      $\mathbf{z}_i(m_t) = \sum_{j\in N_i} P_{ij}\mathbf{z}_j(m_t - 1) + \widehat{\mathbf{g}}_i(m_t - 1)$
      define $F_{m_t,i}(\mathbf{x}) = \eta\mathbf{z}_i(m_t)^\top\mathbf{x} + \|\mathbf{x}\|_2^2$
      $\mathbf{x}_i(m_t) = \text{CGSC}(\mathcal{K}, \epsilon, L, F_{m_t,i}(\mathbf{x}), \mathbf{x}_i(m_t - 1))$
    end if
    play $\mathbf{x}_i(m_t)$ and observe $\mathbf{g}_i(t) = \nabla f_{t,i}(\mathbf{x}_i(m_t))$
  end for
end for

where $\mathcal{T}_m = \{(m-1)K + 1, \cdots, mK\}$.

Initially, we set $\mathbf{x}_i(1) = \mathbf{0} \in \mathcal{K}$ and $\mathbf{z}_i(1) = \mathbf{0}$ for each local learner i. Inspired by (2) and (3) used by Zhang et al. (2017), at the beginning of each block $m > 1$, each local learner i communicates with its neighbors to update $\mathbf{z}_i(m)$ as

$$\mathbf{z}_i(m) = \sum_{j\in N_i} P_{ij}\mathbf{z}_j(m-1) + \widehat{\mathbf{g}}_i(m-1)$$

which is utilized to define $F_{m,i}(\mathbf{x})$ as

$$F_{m,i}(\mathbf{x}) = \eta\mathbf{z}_i(m)^\top\mathbf{x} + \|\mathbf{x}\|_2^2.$$

Similar to the update rules of D-OCG (4), one may simply perform 1 linear optimization step with the above $F_{m,i}(\mathbf{x})$ for each local learner i. However, it will increase the regret, since the number of updates is decreased. To address this problem, we invoke CGSC for each update as

$$\mathbf{x}_i(m) = \text{CGSC}(\mathcal{K}, \epsilon, L, F_{m,i}(\mathbf{x}), \mathbf{x}_i(m-1)).$$

The detailed procedures of our algorithm are presented in Algorithm 2, and it is called distributed block online conditional gradient (D-BOCG). By setting $K = \sqrt{T}$, it is easy to verify that the communication complexity of D-BOCG is only $O(\sqrt{T})$, which is significantly lower than the $O(T)$ complexity of D-OCG. Moreover, we establish the following theorem regarding the regret of each local learner.

Theorem 1 *Let* $\eta = \frac{RT^{-3/4}}{G}$, $\epsilon = 4R^2T^{-1/2}$, $K = T^{1/2}$ *and* $L = \frac{16R^2}{\epsilon^2}(\eta\alpha KG\sqrt{\epsilon} + \eta^2\alpha^2K^2G^2)$, *where* $\alpha = \frac{1+\sigma_2(P)}{1-\sigma_2(P)}\sqrt{n} + 1$. *Under Assumptions 1, 3 and 4, for any* $i \in V$, *Algorithm 2 has*

$$R_{T,i} \leq (8 + 3\alpha')nGRT^{3/4}$$

where $\alpha' = \frac{\sqrt{n}}{1-\sigma_2(P)}$.

It is easy to verify that the total number of linear optimization steps required by our D-BOCG is at most BL. Because of Theorem 1, $B = T/K = \sqrt{T}$ and

$$L = \frac{16R^2}{\epsilon^2}(\eta\alpha KG\sqrt{\epsilon} + \eta^2\alpha^2K^2G^2) = \left(2\alpha + \alpha^2\right)\sqrt{T}$$

our D-BOCG enjoys an $O(T^{3/4})$ regret bound with at most $O(T)$ linear optimization steps, which is on the same order as that required by D-OCG.

3.3. Algorithm for Bandit Setting

To handle the bandit setting, where only the loss value is available to each local learner, the main challenge is due to the lack of gradient. Therefore, we first introduce a standard technique called one-point gradient estimator (Flaxman et al., 2005), which can approximate the gradient with a single loss value.

One-point Gradient Estimator For a function $f(\mathbf{x})$, its δ-smoothed version is defined as

$$\widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{B}^d}[f(\mathbf{x} + \delta\mathbf{u})]$$

and satisfies the following lemma.

Lemma 1 *(Lemma 1 in Flaxman et al. (2005)) Let $\delta > 0$, we have*

$$\nabla\widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{S}^d}\left[\frac{d}{\delta}f(\mathbf{x} + \delta\mathbf{u})\mathbf{u}\right]$$

where $\mathcal{S}^d$ denotes the unit sphere in $\mathbb{R}^d$.

Lemma 1 provides an unbiased estimator of the gradient $\nabla\widehat{f}_\delta(\mathbf{x})$ by only observing the single value $f(\mathbf{x} + \delta\mathbf{u})$.

Combining our D-BOCG with this technique, our algorithm for the bandit setting is outlined in Algorithm 3, and named as distributed block bandit conditional gradient (D-BBCG), where $0 < \delta \leq r$ and

$$\mathcal{K}_\delta = (1 - \delta/r)\mathcal{K} = \{(1 - \delta/r)\mathbf{x}|\mathbf{x} \in \mathcal{K}\}.$$

Compared D-BBCG with D-BOCG, there exist three differences as follows. First, in line 14 of D-BBCG, due to the lack of $\nabla f_{t,i}(\mathbf{x}_i(m_t))$, we adopt the one-point gradient estimator to approximate it as

$$\mathbf{g}_i(t) = \frac{d}{\delta}f_{t,i}(\mathbf{y}_i(t))\mathbf{u}_i(t)$$

where $\mathbf{y}_i(t) = \mathbf{x}_i(m_t) + \delta\mathbf{u}_i(t)$ and $\mathbf{u}_i(t) \sim \mathcal{S}^d$. Second, as in line 13 of D-BBCG, the actual decision $\mathbf{y}_i(t)$ is $\mathbf{x}_i(m_t)$ plus a random decision $\delta\mathbf{u}_i(t)$, which promotes more explorations. Third, to ensure $\mathbf{y}_i(t) \in \mathcal{K}$, in line 10 of D-BBCG, we perform

$$\mathbf{x}_i(m_t) = \text{CGSC}(\mathcal{K}_\delta, \epsilon, L, F_{m_t,i}(\mathbf{x}), \mathbf{x}_i(m_t - 1))$$

Algorithm 3 D-BBCG

1: **Input:** feasible set $\mathcal{K}$, δ, η, L, ϵ and K
2: **Initialization:** choose $\{\mathbf{x}_i(1) = \mathbf{0} \in \mathcal{K}_\delta | i \in V\}$ and set $\{\mathbf{z}_i(1) = \mathbf{0} | i \in V\}$
3: **for** $t = 1, \cdots, T$ **do**
4: $\quad m_t = \lceil t/K \rceil$
5: $\quad$ **for** each local learner $i \in V$ **do**
6: $\quad\quad$ **if** $t > 1$ and $\text{mod}(t, K) = 1$ **then**
7: $\quad\quad\quad \widehat{\mathbf{g}}_i(m_t - 1) = \sum_{k=t-K}^{t-1} \mathbf{g}_i(k)$
8: $\quad\quad\quad \mathbf{z}_i(m_t) = \sum_{j\in N_i} P_{ij}\mathbf{z}_i(m_t - 1) + \widehat{\mathbf{g}}_i(m_t - 1)$
9: $\quad\quad\quad$ define $F_{m_t,i}(\mathbf{x}) = \eta\mathbf{z}_i(m_t)^\top\mathbf{x} + \|\mathbf{x}\|_2^2$
10: $\quad\quad\quad \mathbf{x}_i(m_t) = \text{CGSC}(\mathcal{K}_\delta, \epsilon, L, F_{m_t,i}(\mathbf{x}), \mathbf{x}_i(m_t - 1))$
11: $\quad\quad$ **end if**
12: $\quad\quad \mathbf{u}_i(t) \sim \mathcal{S}^d$
13: $\quad\quad$ play $\mathbf{y}_i(t) = \mathbf{x}_i(m_t) + \delta\mathbf{u}_i(t)$ and observe $f_{t,i}(\mathbf{y}_i(t))$
14: $\quad\quad \mathbf{g}_i(t) = \frac{d}{\delta}f_{t,i}(\mathbf{y}_i(t))\mathbf{u}_i(t)$
15: $\quad$ **end for**
16: **end for**

by replacing $\mathcal{K}$ in line 10 of D-BOCG with a smaller set $\mathcal{K}_\delta \subseteq \mathcal{K}$, which limits $\mathbf{x}_i(m_t)$ in the set $\mathcal{K}_\delta$. Because of Assumption 3 and $0 < \delta \leq r$, it is easy to verify that $\mathbf{x} + \delta\mathbf{u} \in \mathcal{K}$, for any $\mathbf{x} \in \mathcal{K}_\delta$ and $\mathbf{u} \sim \mathcal{S}^d$.

Similar to D-BOCG, the communication complexity of D-BBCG is $O(\sqrt{T})$ by setting $K = \sqrt{T}$, which is also significantly lower than that of D-OCG. Following previous studies for the bandit setting (Flaxman et al., 2005; Garber & Kretzu, 2019), we further assume that the adversary is oblivious (i.e., all local loss functions are chosen beforehand), and establish the following theorem for D-BBCG.

Theorem 2 *Let $\eta = \frac{cR}{\alpha_T dM}T^{-3/4}$, $\delta = cT^{-1/4}$, $\epsilon = 4R^2T^{-1/2}$, $K = T^{1/2}$ and $L = \frac{16R^2}{\epsilon^2}(\eta\alpha\beta\sqrt{\epsilon} + \eta^2\alpha^2\beta^2)$, where $c > 0$ is a constant such that $\delta \leq r$, $\beta = \alpha_T\frac{dM\sqrt{K}}{\delta} + KG$, $\alpha = \frac{1+\sigma_2(P)}{1-\sigma_2(P)}\sqrt{n} + 1$ and $\alpha_T = 1 + \sqrt{8\ln\frac{n\sqrt{T}}{\gamma}}$. Under Assumptions 1, 2, 3 and 4, for any $i \in V$, with probability at least $1 - 2\gamma$, Algorithm 3 has*

$$R_{T,i} \leq O\left(\alpha_T T^{3/4}\right).$$

According to Theorem 2, we first note that our D-BBCG attains a high-probability regret bound of $\widetilde{O}\left(T^{3/4}\right)$, which is almost the same as that of D-BOCG up to a logarithmic factor. Moreover, similar to D-OCG and D-BOCG, our D-BBCG requires at most $O(T)$ linear optimization steps, due to $B = T/K = \sqrt{T}$ and

$$L = \frac{16R^2}{\epsilon^2}(\eta\alpha\beta\sqrt{\epsilon} + \eta^2\alpha^2\beta^2) = \left(2\rho + \rho^2\right)\sqrt{T}$$

where $\rho = \alpha + \frac{\alpha cG}{\alpha_T dM}$.

4. Theoretical Analysis

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

4.1. Proof of Theorem 1

In the beginning, we define several auxiliary variables.

Let $\bar{\mathbf{z}}(m) = \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i(m)$ and $\bar{\mathbf{g}}(m) = \frac{1}{n}\sum_{i=1}^n \widehat{\mathbf{g}}_i(m)$. According to Assumption 4, it is easy to verify that

$$\begin{aligned}\bar{\mathbf{z}}(m+1) =& \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i(m+1)\\ =& \frac{1}{n}\sum_{i=1}^n \left(\sum_{j\in N_i} P_{ij}\mathbf{z}_j(m) + \widehat{\mathbf{g}}_i(m)\right)\\ =& \bar{\mathbf{z}}(m) + \bar{\mathbf{g}}(m).\end{aligned}$$

Then, we define

$$\bar{F}_{m+1}(\mathbf{x}) = \eta\bar{\mathbf{z}}(m+1)^\top\mathbf{x} + \|\mathbf{x}\|_2^2$$

and $\bar{\mathbf{x}}(m+1) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \bar{F}_{m+1}(\mathbf{x})$. Moreover, let $\widehat{\mathbf{x}}_i(m) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \eta\mathbf{z}_i(m)^\top\mathbf{x} + \|\mathbf{x}\|_2^2$ and $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^T f_t(\mathbf{x})$.

Now, we derive an upper bound of $\|\mathbf{x}_i(m) - \bar{\mathbf{x}}(m)\|_2$ with the following three lemmas.

Lemma 2 *(Lemma 6 in Zhang et al. (2017)) Let $\mathbf{z}_i(1) = \mathbf{0}$, $\mathbf{z}_i(m+1) = \sum_{j\in N_i} P_{ij}\mathbf{z}_j(m) + \widehat{\mathbf{g}}_i(m)$ and $\bar{\mathbf{z}}(m) = \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i(m)$ for $m \in [B]$, where P satisfies Assumption 4. For any $i \in V$ and $m \in [B]$, assume $\|\widehat{\mathbf{g}}_i(m)\|_2 \le \beta$, we have*

$$\|\mathbf{z}_i(m) - \bar{\mathbf{z}}(m)\|_2 \le \alpha'\beta$$

where $\alpha' = \frac{\sqrt{n}}{1-\sigma_2(P)}$.

Lemma 3 *(Lemma 5 in Duchi et al. (2011)) Let $\Pi_{\mathcal{K}}(\mathbf{u}, \eta) = \arg\min_{\mathbf{x}\in\mathcal{K}} \eta\mathbf{u}^\top\mathbf{x} + \|\mathbf{x}\|_2^2$. We have*

$$\|\Pi_{\mathcal{K}}(\mathbf{u},\eta) - \Pi_{\mathcal{K}}(\mathbf{v},\eta)\|_2 \le \eta\|\mathbf{u}-\mathbf{v}\|_2.$$

Lemma 4 *Let $\widehat{\mathbf{x}}_i(m) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} F_{m,i}(\mathbf{x})$, for $m \in [B]$. For any $i \in V$ and $m \in [B]$, Algorithm 2 with $\epsilon \le 8R^2$ and $L = \frac{16R^2}{\epsilon^2}(\eta\alpha KG\sqrt{\epsilon} + \eta^2\alpha^2K^2G^2)$ has*

$$F_{m,i}(\mathbf{x}_i(m)) - F_{m,i}(\widehat{\mathbf{x}}_i(m)) \le \epsilon$$

where $\alpha = \frac{1+\sigma_2(P)}{1-\sigma_2(P)}\sqrt{n} + 1$.

According to Algorithm 2 and Assumption 1, we have $\widehat{\mathbf{g}}_i(m) = \sum_{t\in\mathcal{T}_m} \mathbf{g}_i(t)$ and $\|\widehat{\mathbf{g}}_i(m)\|_2 \le KG$, where $\mathcal{T}_m = \{(m-1)K+1, \cdots, mK\}$.

So, applying Lemma 2 with $\|\widehat{\mathbf{g}}_i(m)\|_2 \le KG$, we have

$$\|\mathbf{z}_i(m) - \bar{\mathbf{z}}(m)\|_2 \le \alpha' KG. \tag{6}$$

Then, applying Lemma 3 with (6), we have

$$\|\widehat{\mathbf{x}}_i(m) - \bar{\mathbf{x}}(m)\|_2 \le \eta\|\mathbf{z}_i(m) - \bar{\mathbf{z}}(m)\|_2 \le \eta\alpha' KG$$

which further implies that

$$\begin{aligned}&\|\mathbf{x}_i(m) - \bar{\mathbf{x}}(m)\|_2\\ \le& \|\mathbf{x}_i(m) - \widehat{\mathbf{x}}_i(m)\|_2 + \|\widehat{\mathbf{x}}_i(m) - \bar{\mathbf{x}}(m)\|_2\\ \le& \sqrt{F_{m,i}(\mathbf{x}_i(m)) - F_{m,i}(\widehat{\mathbf{x}}_i(m))} + \eta\alpha' KG\\ \le& \sqrt{\epsilon} + \eta\alpha' KG\end{aligned} \tag{7}$$

where the second inequality is due to the fact that $F_{m,i}(\mathbf{x})$ is 2-strongly convex and (5), and the last inequality is due to Lemma 4.

Let $\epsilon' = \sqrt{\epsilon} + \eta\alpha' KG$. Then, for any $i, j \in V$, we have

$$\begin{aligned}&\sum_{t=1}^T (f_{t,j}(\mathbf{x}_i(m_t)) - f_{t,j}(\mathbf{x}^*))\\ \le& \sum_{t=1}^T (f_{t,j}(\bar{\mathbf{x}}(m_t)) + G\|\bar{\mathbf{x}}(m_t) - \mathbf{x}_i(m_t)\|_2 - f_{t,j}(\mathbf{x}^*))\\ \le& \sum_{t=1}^T (f_{t,j}(\mathbf{x}_j(m_t)) + G\|\bar{\mathbf{x}}(m_t) - \mathbf{x}_j(m_t)\|_2 - f_{t,j}(\mathbf{x}^*))\\ &+ GT\epsilon'\\ \le& \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\mathbf{x}_j(m_t) - \mathbf{x}^*) + 2GT\epsilon'\\ =& \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\mathbf{x}_j(m_t) - \bar{\mathbf{x}}(m_t))\\ &+ \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*) + 2GT\epsilon'\\ \le& \sum_{t=1}^T \|\nabla f_{t,j}(\mathbf{x}_j(m_t))\|_2\|\mathbf{x}_j(m_t) - \bar{\mathbf{x}}(m_t)\|_2\\ &+ \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*) + 2GT\epsilon'\\ \le& \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*)\\ &+ \sum_{t=1}^T G\|\bar{\mathbf{x}}(m_t) - \mathbf{x}_j(m_t)\|_2 + 2GT\epsilon'\\ \le& \sum_{t=1}^T \nabla f_{t,j}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*) + 3GT\epsilon'\end{aligned}$$

where the third inequality is due to the convexity of $f_{t,j}(\mathbf{x})$.

Furthermore, for any $i \in V$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{j=1}^{n} f_{t,j}(\mathbf{x}_i(m_t)) - \sum_{t=1}^{T}\sum_{j=1}^{n} f_{t,j}(\mathbf{x}^*) \\
\leq&\sum_{t=1}^{T}\sum_{j=1}^{n} \nabla f_{t,j,}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*) \\
&+ 3nGT\left(\sqrt{\epsilon} + \eta\alpha' KG\right) \\
=&\sum_{m=1}^{B}\sum_{t\in\mathcal{T}_m}\sum_{j=1}^{n} \nabla f_{t,j,}(\mathbf{x}_j(m_t))^\top(\bar{\mathbf{x}}(m_t) - \mathbf{x}^*) \\
&+ 3nGT\left(\sqrt{\epsilon} + \eta\alpha' KG\right) \\
=&\sum_{m=1}^{B}\sum_{j=1}^{n}\sum_{t\in\mathcal{T}_m} \nabla f_{t,j,}(\mathbf{x}_j(m))^\top(\bar{\mathbf{x}}(m) - \mathbf{x}^*) \\
&+ 3nGT\left(\sqrt{\epsilon} + \eta\alpha' KG\right) \\
=&n\sum_{m=1}^{B} \bar{\mathbf{g}}(m)^\top(\bar{\mathbf{x}}(m) - \mathbf{x}^*) \\
&+ 3nGT\left(\sqrt{\epsilon} + \eta\alpha' KG\right).
\end{aligned} \tag{8}$$

To bound $\sum_{m=1}^{B} \bar{\mathbf{g}}(m)^\top(\bar{\mathbf{x}}(m) - \mathbf{x}^*)$, we introduce the following lemma.

Lemma 5 *(Lemma 2.3 in Shalev-Shwartz (2011)) Let* $\widehat{\mathbf{x}}_t^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) + \mathcal{R}(\mathbf{x}), \forall t \in [T]$*, where* $\mathcal{R}(\mathbf{x})$ *is a strongly convex function. Then,* $\forall \mathbf{x} \in \mathcal{K}$*, it holds that*

$$\begin{aligned}
&\sum_{t=1}^{T}\left(f_t(\widehat{\mathbf{x}}_t^*) - f_t(\mathbf{x})\right) \\
\leq&\mathcal{R}(\mathbf{x}) - \mathcal{R}(\widehat{\mathbf{x}}_1^*) + \sum_{t=1}^{T}\left(f_t(\widehat{\mathbf{x}}_t^*) - f_t(\widehat{\mathbf{x}}_{t+1}^*)\right).
\end{aligned}$$

According to the definition, we have

$$\bar{\mathbf{x}}(m+1) = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \eta\bar{\mathbf{z}}(m+1)^\top\mathbf{x} + \|\mathbf{x}\|_2^2.$$

So, applying Lemma 5 with the linear loss functions $\left\{\bar{\mathbf{g}}(m)^\top\mathbf{x}\right\}_{m=1}^{B}$, the decision set $\mathcal{K}$ and the regularizer $\mathcal{R}(\mathbf{x}) = \frac{\|\mathbf{x}\|_2^2}{\eta}$, we have

$$\begin{aligned}
&\sum_{m=1}^{B} \bar{\mathbf{g}}(m)^\top(\bar{\mathbf{x}}(m) - \mathbf{x}^*) \\
\leq&\frac{\|\mathbf{x}^*\|_2^2}{\eta} - 0 + \sum_{m=1}^{B} \bar{\mathbf{g}}(m)^\top(\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)) \\
\leq&\frac{R^2}{\eta} + \sum_{m=1}^{B} \|\bar{\mathbf{g}}(m)\|_2\|\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)\|_2.
\end{aligned} \tag{9}$$

It is easy to verify that $\bar{F}_{m+1}(\mathbf{x})$ is 2-strongly convex, which implies that

$$\begin{aligned}
&\|\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)\|_2^2 \\
\leq&\bar{F}_{m+1}(\bar{\mathbf{x}}(m)) - \bar{F}_{m+1}(\bar{\mathbf{x}}(m+1)) \\
=&\bar{F}_m(\bar{\mathbf{x}}(m)) + \eta\bar{\mathbf{g}}(m)^\top\bar{\mathbf{x}}(m) \\
&- \bar{F}_m(\bar{\mathbf{x}}(m+1)) - \eta\bar{\mathbf{g}}(m)^\top\bar{\mathbf{x}}(m+1) \\
=&\bar{F}_m(\bar{\mathbf{x}}(m)) - \bar{F}_m(\bar{\mathbf{x}}(m+1)) \\
&+ \eta\bar{\mathbf{g}}(m)^\top\left(\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)\right) \\
\leq&\eta\|\bar{\mathbf{g}}(m)\|_2\|\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)\|_2.
\end{aligned}$$

The above inequality can be simplified as

$$\|\bar{\mathbf{x}}(m) - \bar{\mathbf{x}}(m+1)\|_2 \leq \eta\|\bar{\mathbf{g}}(m)\|_2. \tag{10}$$

Substituting (10) into (9), we have

$$\begin{aligned}
&\sum_{m=1}^{B} \bar{\mathbf{g}}(m)^\top(\bar{\mathbf{x}}(m) - \mathbf{x}^*) \\
\leq&\frac{R^2}{\eta} + \eta\sum_{m=1}^{B} \|\bar{\mathbf{g}}(m)\|_2^2 \\
=&\frac{R^2}{\eta} + \eta\sum_{m=1}^{B} \left\|\frac{1}{n}\sum_{i=1}^{n}\widehat{\mathbf{g}}_i(m)\right\|_2^2 \\
\leq&\frac{R^2}{\eta} + \frac{\eta}{n}\sum_{m=1}^{B}\sum_{i=1}^{n} \|\widehat{\mathbf{g}}_i(m)\|_2^2 \\
=&\frac{R^2}{\eta} + \eta BK^2G^2.
\end{aligned} \tag{11}$$

Finally, substituting (11) into (8), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{j=1}^{n} f_{t,j}(\mathbf{x}_i(m_t)) - \sum_{t=1}^{T}\sum_{j=1}^{n} f_{t,j}(\mathbf{x}^*) \\
\leq&\frac{nR^2}{\eta} + n\eta BK^2G^2 + 3nGT\left(\sqrt{\epsilon} + \eta\alpha' KG\right) \\
=&(8 + 3\alpha')nGRT^{3/4}.
\end{aligned}$$

4.2. Proof of Lemma 4

For brevity, we define $h_m(\mathbf{x}) = F_{m,i}(\mathbf{x}) - F_{m,i}(\hat{\mathbf{x}}_i(m))$ and $h_m = F_{m,i}(\mathbf{x}_i(m)) - F_{m,i}(\hat{\mathbf{x}}_i(m))$.

For $m = 1$, because $\mathbf{x}_i(1) = \hat{\mathbf{x}}_i(1) = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\|_2^2$, we have

$$h_1 = F_{1,i}(\mathbf{x}_i(1)) - F_{1,i}(\hat{\mathbf{x}}_i(1)) = 0 \leq \epsilon. \tag{12}$$

Then, for $m = 2$, we have

$$\begin{aligned}
&h_m(\mathbf{x}_i(m-1)) = F_{m,i}(\mathbf{x}_i(m-1)) - F_{m,i}(\hat{\mathbf{x}}_i(m)) \\
=&F_{m-1,i}(\mathbf{x}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m)) \\
&+ \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top \mathbf{x}_i(m-1) \\
&- \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top \hat{\mathbf{x}}_i(m) \\
\leq&F_{m-1,i}(\mathbf{x}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m-1)) \\
&+ \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top (\mathbf{x}_i(m-1) - \hat{\mathbf{x}}_i(m)) \\
\leq&F_{m-1,i}(\mathbf{x}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m-1)) \\
&+ \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\mathbf{x}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2 \\
\leq&F_{m-1,i}(\mathbf{x}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m-1)) \\
&+ \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\mathbf{x}_i(m-1) - \hat{\mathbf{x}}_i(m-1)\|_2 \\
&+ \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2 \\
\leq&h_{m-1} + \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \sqrt{h_{m-1}} \\
&+ \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2 \\
\leq&\epsilon + \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \sqrt{\epsilon} \\
&+ \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2
\end{aligned} \tag{13}$$

where the first inequality is due to $\hat{\mathbf{x}}_i(m-1) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} F_{m-1,i}(\mathbf{x})$ and the fourth inequality is due to the fact that $F_{m-1}(\mathbf{x})$ is 2-strongly convex and (5).

Moreover, because for each $m = 1, \cdots, B$, $F_{m,i}(\mathbf{x})$ is 2-strongly convex, we also have

$$\begin{aligned}
&\|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2^2 \\
\leq&F_{m,i}(\hat{\mathbf{x}}_i(m-1)) - F_{m,i}(\hat{\mathbf{x}}_i(m)) \\
=&F_{m-1,i}(\hat{\mathbf{x}}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m)) \\
&+ \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top \hat{\mathbf{x}}_i(m-1) \\
&- \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top \hat{\mathbf{x}}_i(m) \\
=&F_{m-1,i}(\hat{\mathbf{x}}_i(m-1)) - F_{m-1,i}(\hat{\mathbf{x}}_i(m)) \\
&+ \eta(\mathbf{z}_i(m) - \mathbf{z}_i(m-1))^\top (\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)) \\
\leq&\eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2 \|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2
\end{aligned}$$

which further implies that

$$\|\hat{\mathbf{x}}_i(m-1) - \hat{\mathbf{x}}_i(m)\|_2 \leq \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2. \tag{14}$$

To bound $\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2$, we introduce the following lemma.

Lemma 6 *(Lemma 3 in Zhang et al. (2017)) Let $\mathbf{z}_i(1) = \mathbf{0}$, $\mathbf{z}_i(m+1) = \sum_{j\in N_i} P_{ij}\mathbf{z}_j(m) + \hat{\mathbf{g}}_i(m)$ and $\bar{\mathbf{z}}(m) = \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i(m)$ for $m \in [B]$, where P satisfies Assumption 4. For any $i \in V$ and $m = 1, \cdots, B$, assume $\|\hat{\mathbf{g}}_i(m)\|_2 \leq \beta$, we have*

$$\|\mathbf{z}_i(m+1) - \mathbf{z}_i(m)\|_2 \leq \alpha\beta$$

where $\alpha = \frac{1+\sigma_2(P)}{1-\sigma_2(P)}\sqrt{n} + 1$.

For $m \in [B]$, applying Lemma 6 with $\|\hat{\mathbf{g}}_i(m)\|_2 \leq KG$, we have

$$\|\mathbf{z}_i(m+1) - \mathbf{z}_i(m)\|_2 \leq \alpha KG. \tag{15}$$

Substituting (14) and (15) into (13), we have

$$\begin{aligned}
&F_{m,i}(\mathbf{x}_i(m-1)) - F_{m,i}(\hat{\mathbf{x}}_i(m)) \\
\leq&\epsilon + \eta\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2\sqrt{\epsilon} \\
&+ \eta^2\|\mathbf{z}_i(m) - \mathbf{z}_i(m-1)\|_2^2 \\
\leq&\epsilon + \eta\alpha KG\sqrt{\epsilon} + \eta^2\alpha^2K^2G^2.
\end{aligned}$$

According to Algorithm 2, we have

$$\mathbf{x}_i(m) = \text{CGSC}(\mathcal{K}, \epsilon, L, F_{m,i}(\mathbf{x}), \mathbf{x}_i(m-1)).$$

To bound $h_m = F_{m,i}(\mathbf{x}_i(m)) - F_{m,i}(\hat{\mathbf{x}}_i(m))$ with the above inequality, we introduce the following lemma regarding the theoretical guarantee of Algorithm 1.

Lemma 7 *(Derived from Lemma 7 of Garber & Kretzu (2019)) Let $F(\mathbf{x}) : \mathcal{K}' \to \mathbb{R}$ be a 2-smooth and 2-strongly convex function, and $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}'} F(\mathbf{x})$, where $\mathcal{K}' \subseteq R\mathcal{B}^d$. Assume $\mathbf{x}_{\text{in}} \in \mathcal{K}'$, $\epsilon \leq 8R^2$ and $L \geq \frac{16R^2}{\epsilon^2}(F(\mathbf{x}_{\text{in}}) - F(\mathbf{x}^*) - \epsilon)$, Algorithm 1 ensures*

$$F(\mathbf{x}_{\text{out}}) - F(\mathbf{x}^*) \leq \epsilon.$$

Because $F_{m,i}(\mathbf{x})$ is 2-smooth and 2-strongly convex, $\epsilon \leq 8R^2$ and $L = \frac{16R^2}{\epsilon^2}(\eta\alpha KG\sqrt{\epsilon} + \eta^2\alpha^2K^2G^2)$, applying Lemma 7 with $\mathcal{K}' = \mathcal{K}$, we have

$$h_m = F_{m,i}(\mathbf{x}_i(m)) - F_{m,i}(\hat{\mathbf{x}}_i(m)) \leq \epsilon$$

for $m = 2$. By induction, we can complete the proof for $m = 1, \cdots, B$.

5. Experiments

In this section, we perform simulation experiments to verify the performance of our proposed algorithms.

5.1. Experimental Settings

Following Zhang et al. (2017), we consider the problem of multiclass classification in the distributed online learning setting. Let k be the number of features, and let h be the number of classes. In the t-th round, each local learner i receives a single example $\mathbf{e}_i(t) \in \mathbb{R}^k$ and chooses a decision matrix $X_i(t) = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \cdots; \mathbf{x}_h^\top] \in \mathbb{R}^{h\times k}$ from the convex set $\mathcal{K} = \{X \in \mathbb{R}^{h\times k} | \|X\|_* \leq \tau\}$, where $\|X\|_*$ denotes the trace norm of X and τ is a constant. Note that $X_i(t)$ can be utilized to predict the class label of $\mathbf{e}_i(t)$ by computing $\operatorname{argmax}_{\ell\in[h]} \mathbf{x}_\ell^\top \mathbf{e}_i(t)$. Then, the true class label $y_i(t)$ is revealed, and it suffers the multivariate logistic loss

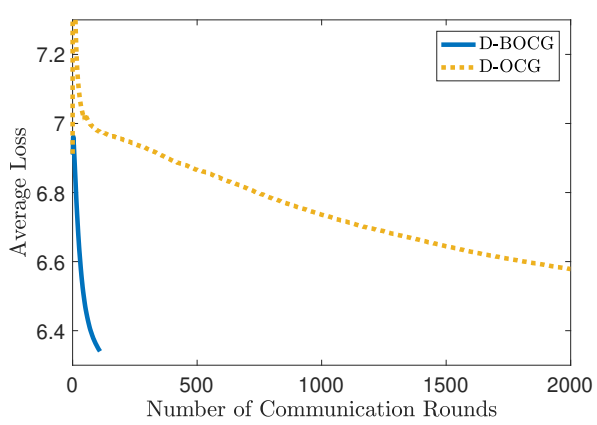

Figure 1. Comparison of D-BOCG and D-OCG on the aloi dataset.

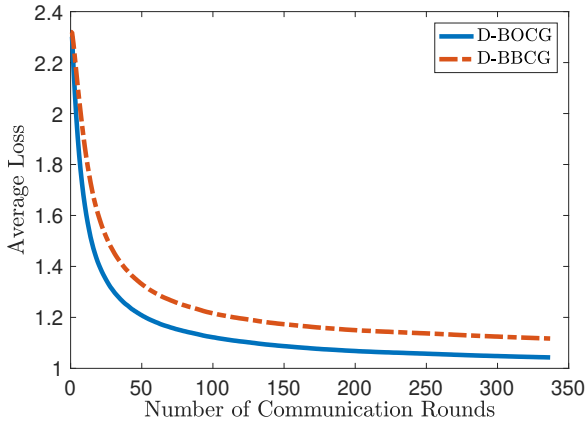

Figure 2. Comparison of our two algorithms on the poker dataset.

$f_{t,i}(X_i(t)) = \log\left(1 + \sum_{\ell \neq y_i(t)} e^{\mathbf{x}_\ell^\top \mathbf{e}_i(t) - \mathbf{x}_{y_i(t)}^\top \mathbf{e}_i(t)}\right)$. The distributed network is modeled by a cycle graph $G = (V, E)$ with 9 nodes, where each node only has three immediate neighbors including itself. The weight matrix P is simply set as $P_{ij} = 1/3$ if $(i, j) \in E$, which satisfies Assumption 4. For any dataset, we will divide it into 9 equally-sized parts, and distribute them onto the computing nodes in the network. To measure the performance of each algorithm, we introduce the average loss defined as $\frac{1}{tn^2} \sum_{q=1}^{t} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{q,j}(X_i(q))$ for the t-th round.

5.2. Experimental Results

To validate the advantage of our D-BOCG on communication complexity, we first compare it against D-OCG (Zhang et al., 2017). As in Zhang et al. (2017), we also use the aloi dataset from the LIBSVM repository (Chang & Lin, 2011), the details of which are summarized in Table 1. According to Zhang et al. (2017), we set the bound of trace norm as $\tau = 50$, and set $s_t = 1/\sqrt{t}$ and $\eta = cT^{-3/4}$ for D-OCG by tunning the constant c. For our D-BOCG, we set $K = \lfloor \sqrt{T} \rfloor$, $\epsilon = 1e-5$, $L = 20$ and $\eta = cT^{-3/4}$ by tunning the constant c. For both D-BOCG and D-OCG, the constant c is selected from $[0.01, \cdots, 1e5]$. Fig. 1 shows the average loss versus the number of communication rounds for D-BOCG and D-OCG. We find that the average loss of our D-BOCG decreases faster than that of D-OCG with the increasing of communication rounds, which verifies the theoretical results of our D-BOCG.

Then, to verify the performance of our D-BBCG, we compare it with our D-BOCG. Note that D-BBCG only uses approximate gradients generated by the one-point gradient estimator, the performance of which is highly affected by the dimensionality. To make a fair comparison, we use the poker dataset from the LIBSVM repository, the dimensionality of which is relatively small. The upper bound of trace norm is set to be $\tau = 1$, and the parameters of D-BOCG are set in the same way as the previous experiment. For D-BBCG, we set $K = \lfloor \sqrt{T} \rfloor$, $\epsilon = 1e-5$, $L = 20$, $\delta = 0.1$ and $\eta = cT^{-3/4}$ by selecting the constant c from $[0.01, \cdots, 1e5]$. Since D-BBCG is a randomized algorithm, we repeat it 10 times and report the average results. Fig. 2 shows the average loss versus the number of communication rounds for D-BOCG and D-BBCG. We find that D-BBCG is worse than D-BOCG, which is reasonable because D-BBCG is working with the more challenging bandit setting.

Table 1. Summary of datasets

Dataset	#Features	#Classes	#Examples
aloi	128	1000	108000
poker	10	10	1025010

6. Conclusion and Future Work

In this paper, we first propose a projection-free algorithm called D-BOCG for distributed online convex optimization, the communication complexity of which is only $O(\sqrt{T})$. According to our analysis, it enjoys an $O(T^{3/4})$ regret bound with at most $O(T)$ linear optimization steps, which matches the best result established by the existing algorithm with $O(T)$ communication complexity. Furthermore, to handle the more challenging bandit setting, we propose our second projection-free algorithm named as D-BBCG, which is a bandit variant of D-BOCG. Similar to D-BOCG, it attains a high-probability regret bound of $\widetilde{O}(T^{3/4})$ with at most $O(T)$ linear optimization steps. An open question is whether the regret bound for the full information setting can be improved if a few projections are allowed. We note that $O(\log T)$ projections are sufficient to achieve the optimal convergence rate for stochastic optimization of smooth and strongly convex functions (Zhang et al., 2013).

Acknowledgements

This work was partially supported by NSFC (61976112, 61921006), and the Fundamental Research Funds for the Central Universities (14380074). The authors would like to thank the anonymous reviewers for their helpful comments.

References

Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 9–16, 2006.

Awerbuch, B. and Kleinberg, R. D. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pp. 45–53, 2004.

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

Azuma, K. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

Blum, A. and Kalai, A. Universal portfolios with and without transaction costs. *Machine Learning*, 35(3):193–205, 1999.

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

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011.

Chen, L., Harshaw, C., Hassani, H., and Karbasi, A. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 814–823, 2018.

Chen, L., Zhang, M., and Karbasi, A. Projection-free bandit convex optimization. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2047–2056, 2019.

Duchi, J. C., Agarwal, A., and Wainwright, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2011.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, 2005.

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

Freund, Y., Schapire, R. E., Singer, Y., and Warmuth, M. K. Using and combining predictors that specialize. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pp. 334–343, 1997.

Garber, D. and Hazan, E. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

Garber, D. and Kretzu, B. Improved regret bounds for projection-free bandit convex optimization. *arXiv:1910.03374*, 2019.

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

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

Hazan, E. and Minasyan, E. Faster projection-free online learning. *arXiv:2001.11568*, 2020.

Hosseini, S., Chapman, A., and Mesbahi, M. Online distributed optimization via dual averaging. In *52nd IEEE Conference on Decision and Control*, pp. 1484–1489, 2013.

Jain, P., Kulis, B., Dhillon, I. S., and Grauman, K. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems 21*, pp. 761–768, 2008.

Levy, K. Y. and Krause, A. Projection free online learning over smooth sets. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1458–1466, 2019.

Li, D., Wong, K. D., Hu, Y. H., and Sayeed, A. M. Detection, classification and tracking of targets in distributed sensor networks. *IEEE Signal Processing Magazine*, 19(2):17–29, 2002.

Nedić, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11): 2506–2517, 2009.

Ram, S. S., Nedić, A., and Veeravalli, V. V. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, 147(3):516–545, 2010.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Tsagkatakis, G. and Savakis, A. Online distance metric learning for object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12):1810–1821, 2011.

Xiao, L., Boyd, S., and Kim, S.-J. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

Zhang, L., Yang, T., Jin, R., and He, X. $O(\log T)$ projections for stochastic optimization of smooth and strongly convex functions. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1121–1129, 2013.

Zhang, W., Zhao, P., Zhu, W., Hoi, S. C. H., and Zhang, T. Projection-free distributed online learning in networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 4054–4062, 2017.

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