Online Boosting with Bandit Feedback

Nataly Brukhim NBRUKHIM@PRINCETON.EDU
Department of Computer Science, Princeton University
and Google AI Princeton

Elad Hazan EHAZAN@PRINCETON.EDU
Department of Computer Science, Princeton University
and Google AI Princeton

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We consider the problem of online boosting for regression tasks, when only limited information is available to the learner. This setting is motivated by applications in reinforcement learning, in which only partial feedback is provided to the learner. We give an efficient regret minimization method that has two implications. First, we describe an online boosting algorithm with noisy multi-point bandit feedback. Next, we give a new projection-free online convex optimization algorithm with stochastic gradient access, that improves state-of-the-art guarantees in terms of efficiency. Our analysis offers a novel way of incorporating stochastic gradient estimators within Frank-Wolfe-type methods, which circumvents the instability encountered when directly applying projection-free optimization to the stochastic setting.

1. Introduction

Boosting is a fundamental methodology in machine learning which allows us to efficiently convert a number of weak learning rules into a strong one. The setting of boosting for batch learning has been studied extensively, leading to a deep and significant theory and celebrated practical success. See (Schapire and Freund, 2012) for a thorough discussion.

In contrast to the batch setting, online learning algorithms typically don't make any stochastic assumptions about the data. They are often faster, memory-efficient, and can adapt to the best changing predictor over time. A line of previous work has explored extensions of boosting methods to the online learning setting (Leistner et al., 2009; Chen et al., 2012, 2014; Beygelzimer et al., 2015b,a; Agarwal et al., 2019; Brukhim et al., 2020). Of these, several works (Beygelzimer et al., 2015a; Agarwal et al., 2019) formally address the setting of online boosting for regression, providing theoretical guarantees on variants of the Gradient boosting method (Friedman, 2001; Mason et al., 2000) widely used in practice. However, such guarantees are only provided under the assumption that full information is available to the learner, i.e., that the entire loss function is revealed after each prediction is made.

On the other hand, in many online learning problems, the feedback available to the learner is limited. These problems naturally occur in many practical applications, in which interactions with the environment are costly, and the learner has to operate under bandit feedback. Such is often the case, for example, for Reinforcement Learning in a Markov decision process (Jin and Luo, 2019; Rosenberg and Mansour, 2019b). In the bandit feedback model, the learner only observes the loss values related to predictions she chose. In particular, the loss function is not revealed to the learner

and, unless the prediction was correct, the true label remains unknown. In this paper we propose the first online boosting algorithm with theoretical guarantees, in the bandit feedback setting.

The underlying ideas used in our approach are based on the fact that boosting can be seen as an optimization procedure. It can be interpreted as cost minimization over the set of linear combinations of weak learners. That is, boosting can be thought of as applying a gradient-descent-type algorithm in a function space (Schapire and Freund, 2012; Friedman, 2001; Mason et al., 2000). This functional view of boosting has also inspired a few studies of boosting methods (Friedman, 2001; Wang et al., 2015; Beygelzimer et al., 2015a) that are based on the classical Frank-Wolfe projection free optimization algorithm (Frank and Wolfe, 1956).

In this work we leverage these ideas to yield a new online boosting algorithm based on a Frank-Wolfe-type technique. Namely, our online boosting algorithm is based on a projection-free Online Convex Optimization (OCO) method with stochastic gradients. The stochastic gradient assumption can capture, in particular, bandit feedback, since stochastic gradient estimates can be obtained using random function evaluation (Flaxman et al., 2005).

However, such existing projection-free OCO methods either achieve suboptimal regret bounds (Hazan and Kale, 2012) or have high per-iteration computational costs (Mokhtari et al., 2018; Chen et al., 2018; Xie et al., 2019). To fill this gap, we derive a new method and analysis of a projection-free OCO algorithm with stochastic gradients. As summarized in Table 1, our projection-free OCO algorithm is the fastest known method compared to previous work, while achieving an optimal regret bound. Furthermore, our Frank-Wolfe-type algorithm gives rise to an efficient online boosting method in the bandit setting.

Our results We propose new online learning methods using only limited feedback. Specifically:

- **Online Boosting with Bandit Feedback**, we propose the first online boosting algorithm with theoretical regret bounds in the bandit feedback setting. The formal description of our method is given in Algorithm 2, and its theoretical guarantees are stated in Theorem 11. In addition, Section D in the Appendix presents encouraging experiments on benchmark datasets.

- **Projection-Free OCO with Stochastic Gradients**, an efficient projection-free OCO algorithm, with stochastic gradients, which improves the best known guarantees in terms of computational efficiency. Table 1 compares these results to previous work. Our method is given in Algorithm 1, and its theoretical guarantees are stated in Theorems 2 and 3.

Algorithm	Regret	Per-round Cost	Feedback	Guarantee
Online-FW (Hazan and Kale, 2012)	$O(T^{3/4})$	$O(1)$	Full	deterministic
Meta-FW (Chen et al., 2018)	$O(\sqrt{T})$	$O(T^{3/2})$	Stochastic	in expectation
MORGFW (Xie et al., 2019)	$\tilde{O}(\sqrt{T})$	$O(T)$	Stochastic	w.h.p.
This Work (Thm. 3)	$\tilde{O}(\sqrt{T})$	$O(\sqrt{T})$	Stochastic	w.h.p.

Table 1: Comparison of projection-free Online Convex Optimization methods.

1.1. Techniques and challenges

The main challenge in designing a projection-free OCO method in the partial information setting is the fact that Frank-Wolfe type approaches are provably not robust to stochastic gradients (see e.g. Hassani et al. (2017), Appendix B, for an example and further discussion on why Frank-Wolfe type methods do not easily admit stochastic variants). Thus, the straightforward approach of using unbiased gradient estimators in conjunction with Frank-Wolfe does not apply. Previous works (Mokhtari et al., 2018; Chen et al., 2018; Xie et al., 2019) have mitigated these issues by applying intricate variance reduction techniques, that take a toll on computational efficiency.

An important observation given in this paper is that when sequentially running multiple Frank-Wolfe procedures within an OCO framework, such issues are entirely eliminated. Our analysis demonstrates that the stochasticity of the gradients is conditionally independent of other sources of randomness in the algorithm. This enables the derivation of a simple and efficient projection-free OCO method in the stochastic setting.

As the main goal of this work is to provide an online boosting framework for the bandit setting, we employ gradient estimation techniques (Flaxman et al., 2005) that enable to remove the assumption of stochastic gradient oracle access, and thus apply to the more general bandit setting. In our analysis, we detail how our approach applies to the multi-point *noisy* bandit setting, where we make no distributional assumptions on the noise apart from the fact that it is zero-mean and bounded.

Lastly, we derive an effective online boosting framework, which converts a γ-approximate (weak) online learner for linear loss functions that expect full information, to a 1-competitive (strong) online learner for any sequence of convex loss functions, which applies in the noisy multi-point bandit setting.

Paper outline. In the next subsection we discuss related work. Section 2 deals with the setting of projection-free online convex optimization, with stochastic gradient oracle. We describe the OCO algorithm and formally state its theoretical guarantees, followed by the analysis and proofs of our results. In Section 3 we describe a generalization of these techniques, give our main algorithm of online boosting in the bandit feedback model, and formally state its theoretical guarantees. In Section 4 we conclude our results and discuss future work. The complete proofs of our results are given in the Appendix, as well as an empirical evaluation of our boosting algorithm (see Section D).

1.2. Related work

Projection-free OCO. The classical Frank-Wolfe (FW) method was introduced in (Frank and Wolfe, 1956) for efficiently solving linear programming. The framework of Online Convex Optimization (OCO) was introduced by (Zinkevich, 2003), with the online projected gradient descent method, achieving $O(\sqrt{T})$ regret bound. However, the projections required for such an algorithm are too expensive for many large-scale online problems. The online variant of the FW algorithm that applies to general OCO was given in (Hazan and Kale, 2012). It attains $O(T^{3/4})$ regret for the general OCO setting, with only one linear optimization step per iteration. Recent work (Hazan and Minasyan, 2020) on projection-free OCO proposes an approach which guarantees an improved regret bound of $O(T^{2/3})$. A more general setting considers the use of stochastic gradient estimates instead of exact gradients (Mokhtari et al., 2018; Chen et al., 2018; Xie et al., 2019). Although it enables to remove the assumption that exact gradient computation is tractable, it often requires larger computational costs per-iteration. In this work, we give a projection-free OCO method that improves state-of-the-art guarantees with $O(\sqrt{T})$ regret bound, and $O(\sqrt{T})$ per-round cost.

Online Boosting Previous works on online boosting have mostly focused on classification tasks (Leistner et al., 2009; Chen et al., 2012, 2014; Beygelzimer et al., 2015b; Jung et al., 2017; Jung and Tewari, 2018). The main result in this paper is a generalization of the online boosting for regression problems by (Beygelzimer et al., 2015a), to the bandit feedback model. We combine these ideas with zero-order convex optimization techniques (Flaxman et al., 2005), and with our novel projection-free OCO algorithm and analysis. Recent works have also considered online boosting in the bandit setting for classification tasks (Chen et al., 2014; Zhang et al., 2018). These works give convergence guarantees in the more restricted mistake-bound model, whereas in this work we provide regret bounds, compared to a reference function class. The related works of (Garber, 2017; Hazan et al., 2018) consider the metric of α-regret, which is applicable to computationally-hard problems.

Multi-Point Bandit Feedback In this work we consider a relaxation of the standard bandit setting: noisy multi-point bandit feedback. In this model, the learner can query each loss function at multiple points, and obtains noisy feedback values. This model is motivated by reinforcement learning in Markov decision processes, as well as problems in submodular maximization (see discussion section). Previous work on the multi-point bandit model allows multi-point *noiseless* feedback (Agarwal et al., 2010; Duchi et al., 2015; Shamir, 2017). Noiseless feedback is significantly less challenging, since with only two points one can get an arbitrarily good approximation to the gradient. In addition, other works have also considered a *single point* projection-free noiseless bandit model (Garber and Kretzu, 2019; Chen et al., 2019).

2. Projection-Free OCO with Limited Feedback

Consider the setting of Online Convex Optimization (OCO), when only limited feedback is available to the learner, rather than full information. Recall that in the OCO framework (see e.g. (Hazan, 2016)), an online player iteratively makes decisions from a compact convex set $\mathcal{K} \subset \mathbb{R}^d$. At iteration $t = 1, ..., T$, the online player chooses $x_t \in \mathcal{K}$, and the adversary reveals the cost ℓ_t, chosen from $\mathcal{L}$ a family of bounded convex functions over $\mathcal{K}$. The metric of performance in this setting is regret: the difference between the total loss of the learner and that of the best fixed decision in hindsight. Formally, the regret of the OCO algorithm is defined by:

$$R_{\mathcal{A}}^{\mathcal{L}}(T) = \sum_{t=1}^{T} \ell_t(x_t) - \inf_{x^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(x^*). \tag{1}$$

In this work we restrict the information that the learner has with respect to the loss function ℓ_t. Specifically, we focus on two such types of limited feedback:

1. Stochastic Gradients: the learner is only provided with stochastic gradient estimates.
2. Bandit Feedback: the learner only observes the loss values of predictions she made.

Our goal is to design an algorithm which has low regret and low cost per iteration t. We begin with the more restricted setting which assumes access to a stochastic gradient oracle. In Section 3.2 we describe a reduction for the more general bandit setting, in the context of online boosting.

As in previous methods of projection-free OCO (Mokhtari et al., 2018; Chen et al., 2018; Xie et al., 2019), we assume oracle access to an Online Linear Optimizer (OLO). The OLO algorithm optimizes linear objectives in a sequential manner, and has sublinear regret guarantees. A formal definition is given below.

Definition 1 *Let $\mathcal{L}'$ denote a class of linear loss functions, $\ell' : \mathcal{K} \to \mathbb{R}$, with σ-bounded gradient norm (i.e., $\|\nabla \ell'(x)\| \leq \sigma$). An algorithm $\mathcal{A}$ is an **Online Linear Optimizer (OLO)** for $\mathcal{K}$ w.r.t. $\mathcal{L}'$, if for any sequence $\ell'_1, ..., \ell'_T \in \mathcal{L}'$, the algorithm has expected regret w.r.t. $\mathcal{L}'$, $\mathbb{E}[R_{\mathcal{A}}(T, \sigma)]$[1] that is sublinear in T, where expectation is taken w.r.t the internal randomness of $\mathcal{A}$.*

Suitable choices for an OLO include Follow the Perturbed Leader (Kalai and Vempala, 2005), Online Gradient Descent (Zinkevich, 2003), Regularized Follow The Leader (Hazan, 2016), etc.

Denote the diameter of the set $\mathcal{K}$ by $D > 0$, (i.e., $\forall x, x' \in \mathcal{K}$, $\|x - x'\| \leq D$), denote by $G > 0$ an upper bound on the norm of the gradients of $\ell \in \mathcal{L}$ over $\mathcal{K}$ (i.e., $\forall \ell \in \mathcal{L}, x \in \mathcal{K}, \|\nabla \ell(x)\| \leq G$), and denote by $M > 0$ an upper bound on the loss (i.e., $\forall \ell \in \mathcal{L}, x \in \mathcal{K}, |\ell(x)| \leq M$). We also make the following common assumptions:

Assumption 1 *The loss functions $\ell \in \mathcal{L}$ are β-smooth, i.e., for any $x, x' \in \mathcal{K}$, $\ell \in \mathcal{L}$,*

$$\|\nabla \ell(x) - \nabla \ell(x')\| \leq \beta \|x - x'\|.$$

Assumption 2 *The stochastic gradient oracle $\mathcal{O}$ returns an unbiased estimate $\mathbf{g}_t = \mathcal{O}(x, t)$, for any $t \in [T], x \in \mathcal{K}$, and with bounded norm, i.e.,*

$$\mathbb{E}[\mathbf{g}_t] = \nabla \ell_t(x) \ , \ \|\mathbf{g}_t\|^2 \leq \sigma^2.$$

2.1. Algorithm and Analysis

At a high level, our algorithm maintains oracle access to N copies of an OLO algorithm, and iteratively produces points x_t by running a subroutine of a N-step Frank-Wolfe procedure. It uses previous OLOs' predictions, and gradient estimates oracle in place of exact optimization with true gradients. To update parameters, at each iteration t, the algorithm queries the gradient oracle $\mathcal{O}$ at N points. Then, the gradient estimates are fed to the N OLO oracles as linear loss functions. Intuitively, it guides each OLO algorithm to correct for mistakes of the preceding OLOs. A formal description is provided in Algorithm 1.

Algorithm 1 Projection-Free OCO with Stochastic Gradients Oracle

Oracle access: OLO algorithms $\mathcal{A}_1$,...,$\mathcal{A}_N$ (Definition 1), and a stochastic gradient oracle $\mathcal{O}$.
Set step length $\eta_i = \frac{2}{i+1}$ for $i \in [N]$.
for $t = 1, \ldots, T$ **do**
  Define $x_t^0 = \mathbf{0}$.
  for $i = 1$ to N **do**
    Define $\mathbf{x}_t^i = (1 - \eta_i)\mathbf{x}_t^{i-1} + \eta_i \mathcal{A}_i(\mathbf{g}_{1,i}, \ldots, \mathbf{g}_{t-1,i})$.
    Receive stochastic gradient feedback $\mathbf{g}_{t,i} = \mathcal{O}(\mathbf{x}_t^{i-1})$, such that $\mathbb{E}[\mathbf{g}_{t,i}] = \nabla \ell_t(\mathbf{x}_t^{i-1})$.
    Define linear loss function $\ell_t^i(x) = \mathbf{g}_{t,i}^\top \cdot x$, and pass it to OLO $\mathcal{A}_i$.
  end for
  Output prediction $x_t := \mathbf{x}_t^N$.
  Receive loss value $\ell_t(x_t)$.
end for

The following Theorem states the regret guarantees of Algorithm 1. In this paper, all bounds are given with respect to the dependence on the different parameters, and omit all constants.

1. For ease of presentation we denote $R_{\mathcal{A}}(T, \sigma) := R_{\mathcal{A}}^{\mathcal{L}'}(T)$.

Theorem 2 *Given that assumptions 1 - 2 hold, then Algorithm 1 is a projection-free OCO algorithm which only requires* $N = \frac{\beta D}{\sigma}\sqrt{T}$ *stochastic gradient oracle calls per iteration, such that for any sequence of convex losses* $\ell_t \in \mathcal{L}$*, and any* $x^* \in \mathcal{K}$*, its expected regret is,*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x^*)\right] \leq O\left(\sigma D\sqrt{T}\right).$$

The theoretical guarantees given in Theorem 2 use expected regret as the performance metric. Even though expected regret is a widely accepted metric for online randomized algorithms, one might want to rule out the possibility that the regret has high variance, and verify that the given result actually holds with high probability. By observing that excess loss can be formulated as a martingale difference sequence, and by applying analysis using the Azuma-Hoeffding inequality, we can obtain regret guarantees which hold with high probability. The main result is stated below.

Theorem 3 *Given that assumptions 1 - 2 hold, then Algorithm 1 is a projection-free OCO algorithm which only requires* $N = \frac{\beta D}{\sigma}\sqrt{T}$ *stochastic gradient oracle calls per iteration, such that for any* $\rho \in (0,1)$*, and any sequence of convex losses* $\ell_t \in \mathcal{L}$ *over convex set* $\mathcal{K}$*, w.p. at least* $1-\rho$*,*

$$\sum_{t=1}^{T} \ell_t(x_t) - \inf_{x^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(x^*) \leq O\left(\sigma D\sqrt{T \log \frac{\beta D T}{\sigma \rho}}\right).$$

The complete analysis and proofs of the theorems is deferred to the Appendix. Below we give an overview of the main ideas used in the proof of Theorem 2. For simplicity assume an oblivious adversary (although using a standard reduction, our results can be generalized to an adaptive one) [2].

Let $\ell_1, ..., \ell_T$ be any sequence of losses in $\mathcal{L}$. Observe that the only sources of randomness at play are: the OLOs' ($\mathcal{A}_i$'s) internal randomness, and the stochasiticity of the gradients. The analysis below is given in expectation with respect to all these random variables. Note the following fact used in the analysis; for any t, i, the random variables $\mathbf{g}_{t,i}$ and $\mathcal{A}_i(\mathbf{g}_{1,i}, \ldots, \mathbf{g}_{t-1,i})$ (i.e., the output of $\mathcal{A}_i$ at time t) are conditionally independent, given all history up to time t and step $i-1$. This fact allows to derive the following Lemma:

Lemma 4 *For any* $t \in [T]$ *and* $i \in [N]$*, let* $\mathbf{g}_{t,i}$ *be the unbiased stochastic gradient estimate used in Algorithm 1. Denote the output of algorithm* $\mathcal{A}_i$ *at time* t *as* $x_{t,i}$*. Then, we have,*

$$\mathbb{E}\left[\ell_t^i(x_{t,i})\right] = \mathbb{E}\left[\nabla \ell_t(\mathbf{x}_t^{i-1})^\top \cdot x_{t,i}\right].$$

2. See discussion in (Cesa-Bianchi and Lugosi, 2006), Pg. 69, as well as Exercise 4.1 formulating the reduction.

Proof

$$\begin{aligned}
\mathbb{E}\big[\ell_t^i(x_{t,i})\big] &= \mathbb{E}\big[\mathbf{g}_{t,i}^\top \cdot x_{t,i}\big] && \text{(definition of } \ell_t^i(\cdot)\text{)} \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\Big[\mathbb{E}\big[\mathbf{g}_{t,i}^\top \cdot x_{t,i}\big|\mathcal{I}_t^{i-1}\big]\Big] && \text{(law of total expectation)} \\
&\text{(} \mathcal{I}_t^{i-1} \text{ denotes the } \sigma\text{-algebra measuring all sources of randomness up to time } t, i-1.\text{)} \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\Big[\mathbb{E}_{\mathbf{g}_{t,i}}\big[\mathbf{g}_{t,i}\big|\mathcal{I}_t^{i-1}\big]^\top \cdot \mathbb{E}_{\mathcal{A}_i}\big[x_{t,i}\big|\mathcal{I}_t^{i-1}\big]\Big] && \text{(conditional independence)} \\
&\text{(Inner expectations are w.r.t gradient stochasiticity,} \\
&\qquad \text{and } A_i\text{'s internal randomness, respectively.)} \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\Big[\nabla\ell_t(\mathbf{x}_t^{i-1})^\top \cdot \mathbb{E}\big[x_{t,i}\big|\mathcal{I}_t^{i-1}\big]\Big] && \text{(Since } \mathbb{E}[\mathbf{g}_{t,i}] = \nabla\ell_t(\mathbf{x}_t^{i-1})\text{)} \\
&= \mathbb{E}\big[\nabla\ell_t(\mathbf{x}_t^{i-1})^\top \cdot x_{t,i}\big]
\end{aligned}$$

■

Using Lemma 4, the algorithm is analyzed along the lines of the Frank-Wolfe algorithm, obtaining the expected regret bound of Algorithm 1.

Proposition 5 *Given that assumptions 1 - 2 hold, and given oracle access to N copies of an OLO algorithm for linear losses, with $R_{\mathcal{A}}(T,\sigma)$ regret (see Definition 1), Algorithm 1 is an online learning algorithm, such that for any sequence of convex losses $\ell_t \in \mathcal{L}$, and any $x^* \in \mathcal{K}$, its expected regret is,*

$$\mathbb{E}\left[\sum_{t=1}^T \ell_t(x_t) - \sum_{t=1}^T \ell_t(x^*)\right] \leq \frac{2\beta D^2 T}{N} + R_{\mathcal{A}}(T,\sigma).$$

Proof Let $x_{t,i} \in \mathcal{K}$ be the output of the OLO algorithm $\mathcal{A}_i$ at time t, and let x^* be any $\in \mathcal{K}$. The regret definition of $\mathcal{A}_i$ (Definition 1), and the definition of $\ell_t^i(\cdot)$ in Algorithm 1, imply that:

$$\mathbb{E}\left[\sum_{t=1}^T \mathbf{g}_{t,i}^\top \cdot x_{t,i} - \sum_{t=1}^T \mathbf{g}_{t,i}^\top \cdot x^*\right] \leq R_{\mathcal{A}}(T). \tag{2}$$

By applying Lemma 6, we have,

$$\Delta_i \leq (1-\eta_i)\Delta_{i-1} + \frac{\eta_i^2 \beta D^2}{2} T + \eta_i \sum_{t=1}^T \Big(\mathbf{g}_{t,i}^\top (x_{t,i} - x^*) + \boldsymbol{\zeta}_{t,i}\Big)$$

where $\Delta_i \triangleq \sum_{t=1}^T \ell_t(\mathbf{x}_t^i) - \ell_t(x^*)$, and $\boldsymbol{\zeta}_{t,i} \triangleq (\nabla\ell_t(\mathbf{x}_t^{i-1}) - \mathbf{g}_{t,i})^\top \cdot (x_{t,i} - x^*)$, for $i \in [N]$. Take expectation on both sides. By Lemma 4, we have $\mathbb{E}[\boldsymbol{\zeta}_{t,i}] = 0$, and by the OLO guarantee (2), we get that,

$$\mathbb{E}\big[\Delta_i\big] \leq (1-\eta_i)\mathbb{E}\big[\Delta_{i-1}\big] + \frac{\eta_i^2 \beta D^2}{2} T + \eta_i R_{\mathcal{A}}(T)$$

By Claim 7, we get for all $i > 0$ that,

$$\mathbb{E}\big[\Delta_i\big] \leq \frac{2\beta D^2 T}{i+1} + R_{\mathcal{A}}(T). \tag{3}$$

Applying the bound in Equation (3) for $i = N$ concludes the proof. ■

2.2. Proof of Theorem 2

Proof The proof of Theorem 2 is a direct Corollary of Proposition 5, by plugging *Follow the Perturbed Leader* (Kalai and Vempala, 2005) as the OLO algorithm required for Algorithm 1. We get that the regret of the base algorithms $\mathcal{A}_i$ is $R_{\mathcal{A}}(T, \sigma) = O(\sigma D\sqrt{T})$ w.r.t the sequence of linear losses $\{\ell_t^i\}_t$, where D is the diameter of the set $\mathcal{K}$, and σ is the stochastic gradient norm bound (Assumption 2). Thus, by setting $N = \frac{\beta D}{\sigma}\sqrt{T}$, we get expected regret of $O(\sigma D\sqrt{T})$ w.r.t the convex loss sequence $\{\ell_t\}_t$. ■

It remains to state the following technical Lemmas that are used in the main analysis of Algorithm 1 and are the core part that is based on the Frank-Wolfe technique. These Lemmas are used in the proof of Proposition 5 above, as well as in the analysis of our method in the boosting setting, detailed in the following sections. Their proofs are deferred to the Appendix.

Lemma 6 *Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be any convex, β-smooth function. Let $\mathcal{Z} \subset \mathbb{R}^d$ be a set of points with bounded diameter D. Let $i \in \mathbb{N}$, and let $z_1, ..., z_i \in \mathcal{Z}$. Let $\eta_i \in (0, 1)$, and $\gamma \geq 1$. Define,*

$$z^i = (1 - \eta_i)z^{i-1} - \frac{\eta_i}{\gamma} z_i,$$

and g_i a random variable, such that $\mathbb{E}[g_i] = \nabla\ell(z^{i-1})$. Denote $\zeta_i = (\nabla\ell(z^{i-1}) - g_i)^\top(\frac{1}{\gamma}z_i - z)$. Then, for any $z \in \mathcal{Z}$,

$$\Big(\ell(z^i) - \ell(z)\Big) \;\leq\; (1 - \eta_i)\Big(\ell(z^{i-1}) - \ell(z)\Big) + \eta_i\Big(g_i^\top(\frac{1}{\gamma}z_i - z) + \frac{\eta_i\beta D^2}{2\gamma^2} + \zeta_i\Big).$$

Claim 7 *Define $\eta_i = 2/(i+1)$, for some $i \in \mathbb{N}$. Let $C_1, C_2 > 0$ be some constants, and define $\phi_i \in \mathbb{R}$, such that,*

$$\phi_i \;\leq (1 - \eta_i)\phi_{i-1} + \frac{\eta_i^2 C_1}{2} + \eta_i C_2.$$

Then, it holds that $\phi_i \;\leq \eta_i C_1 + C_2$.

3. Online Boosting with Bandit Feedback

The projection-free OCO method given in Section 2, assumes oracle access to an online linear optimizer (OLO), and utilizes it by iteratively making oracle calls with modified objectives, in order to solve the harder task of convex optimization. Analogously, boosting algorithms typically assume oracle access to a "weak" learner, which are utilized by iteratively making oracle calls with modified objective, in order to obtain a "strong" learner, with boosted performance. In this section, we derive an online boosting method in the bandit setting, based on an adaptation of Algorithm 1.

In the online learning setting, we assume that in each round t for $t = 1, 2, ..., T$, an adversary selects an example $x_t \in \mathcal{X}$ and a loss function $\ell_t : \mathcal{Y} \to \mathbb{R}$, where $\mathcal{Y} \subset \mathbb{R}^d$. The loss ℓ_t is chosen from a class of bounded convex losses $\mathcal{L}$. The adversary then presents x_t to the online learning algorithm $\mathcal{A}$, which predicts $\mathcal{A}(x_t)$ in the goal of minimizing the sum of losses over time, when compared against a function class $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$. Specifically, the metric of performance in this setting is policy regret: the difference between the total loss of the learner's predictions, and that of the best fixed policy/function $f \in \mathcal{F}$, in hindsight:

$$R_{\mathcal{A}}^{\mathcal{L}}(T) = \sum_{t=1}^{T} \ell_t(\mathcal{A}(x_t)) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell_t(f(x_t)). \tag{4}$$

To compare this setting with the OCO setting detailed in Section 2, observe that in the OCO setting, at every time step, the adversary only picks the loss function, and the online player picks a point in the decision set $\mathcal{K}$, towards minimizing the loss and competing with the best fixed point in hindsight. On the other hand, in this online learning setting, at every time step the adversary picks both an example and a loss function, and the online player picks a point in $\mathcal{Y}$, towards minimizing the loss and competing with the best fixed mapping in hindsight, of examples in $\mathcal{X}$ to labels in $\mathcal{Y}$. Considering these observations, we describe the online boosting methodology next.

Generalizing from the offline setting for boosting, the notion of a weak learning algorithm is modeled as an online learning algorithm for linear loss functions that competes with a base class of regression functions, while a strong learning algorithm is an online learning algorithm with convex loss functions that competes with a larger class of regression functions. We follow a similar setting to that of the full information Online Gradient Boosting method Beygelzimer et al. (2015a), in the more general case of noisy, bandit feedback, and a weaker notion of weak learner.

Definition 8 *Let $\mathcal{F}$ denote a reference class of regression functions $f : \mathcal{X} \to \mathcal{Y}$, let T denote the horizon length, and let $\gamma \geq 1$ denote the advantage. Let $\mathcal{L}'$ denote a class of linear loss functions, $\ell' : \mathcal{Y} \to \mathbb{R}$. An online learning algorithm $\mathcal{A}$ is a (γ, T)-**agnostic weak online learner (AWOL)** for $\mathcal{F}$ w.r.t. $\mathcal{L}'$, if for any sequence $(x_1, \ell'_1), ..., (x_T, \ell'_T) \in \mathcal{X} \times \mathcal{L}'$, at every iteration $t \in [T]$, the algorithm outputs $\mathcal{A}(x_t) \in \mathcal{Y}$ such that for any $f \in \mathcal{F}$,*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell'_t\big(\mathcal{A}(x_t)\big) - \gamma \sum_{t=1}^{T} \ell'_t\big(f(x_t)\big)\right] \leq R_{\mathcal{A}}(T, \sigma),$$

where the expectation is taken w.r.t the randomness of the weak learner $\mathcal{A}$ and that of the adversary, and the regret $R_{\mathcal{A}}(T, \sigma)$ is sub-linear in T.

Note the slight abuse of notation here; $\mathcal{A}(\cdot)$ is not a function but rather the output of the online learning algorithm $\mathcal{A}$ computed on the given example using its internal state. Observe that the above definition is the natural extension of the γ-approximation guarantee of a standard classification weak learner in the statistical setting Schapire and Freund (2012), to regression tasks in online learning.

The weak learning algorithm is "weak" in the sense that it is only required to, (a) learn linear loss functions, (b) succeed on full-information feedback, and (c) γ-approximate the best predictor in its reference class $\mathcal{F}$, up to an additive regret. Our main result is an online boosting algorithm (Algorithm 2) that converts a weak online learning algorithm, as defined above, into a strong online learning algorithm. The resulting algorithm is "strong" in the sense that it, (a) learns convex loss functions, (b) relies on bandit feedback only, and (c) 1-approximates the best predictor in a larger class of functions, $\text{CH}(\mathcal{F})$ the convex hull of the base class $\mathcal{F}$, up to an additive regret.

3.1. Setting

At every round t, the learner predicts $y \in \mathcal{Y}$, and receives the noisy bandit feedback $\tilde{\ell}_t(y) = \ell_t(y) + w$, where the noise is drawn i.i.d from a distribution $\mathcal{D}$. We make no distributional assumptions on the noise apart from the fact that it is zero-mean and bounded. Denote the diameter of the set $\mathcal{Y}$ by $D > 0$, (i.e., $\forall y, y' \in \mathcal{Y}$, $\|y - y'\| \leq D$), denote by $L > 0$ an upper bound on the norm of the gradients of $\ell \in \mathcal{L}$ over $\mathcal{X}$ (i.e., $\forall \ell \in \mathcal{L}, x \in K, \|\nabla \ell(x)\| \leq L$), and denote by $M > 0$ an upper bound on the loss (i.e., $\forall \ell \in \mathcal{L}, y \in \mathcal{Y}, |\ell(x)| \leq M$). Denote the bound on the noise by M w.l.o.g. (i.e., $|w| \leq M$ for all $w \sim \mathcal{D}$). Additionally, assume that the set $\mathcal{Y}$ is endowed with a projection operation, that we denote by $\Pi_{\mathcal{Y}}$, and satisfies the following properties,

Assumption 3 *The function $\Pi_{\mathcal{Y}} : \mathbb{R}^d \mapsto \mathcal{Y}$ satisfies that for any $z \in \mathbb{R}^d$, $\ell \in \mathcal{L}$, $\ell\big(\Pi_{\mathcal{Y}}(z)\big) \leq \ell(z)$.*

Consider the following example which demonstrates that Assumption 3 is in fact a realistic assumption: for any $\mathcal{Y} \subset \mathbb{R}^d$ let the class of loss functions $\mathcal{L}$ contain losses that are of the form $\ell(y) = \|y - y_t\|^2$ for some $y_t \in \mathcal{Y}$, and let $\Pi_{\mathcal{Y}}(z) \triangleq \arg\min_{y \in \mathcal{Y}} \|z - y\|$ be the Euclidean projection. Indeed, it can be shown that for any $z \in \mathbb{R}^d$, $\|\Pi_{\mathcal{Y}}(z) - y_t\|^2 \leq \|z - y_t\|^2$, simply by a generalization of the Pythagorean theorem. [3]

3.2. Stochastic Gradients to Bandit Feedback

We build on the techniques shown in Section 2, and describe an implementation of the unbiased stochastic gradient oracle, in the bandit setting. Recall that in the bandit feedback model, the only information revealed to the learner at iteration t is the loss $\ell_t(x_t)$ at the point x_t that she has chosen. In particular, the learner does not know the loss had she chosen a different point x_t.

We consider a more relaxed noisy multi-point bandit setting, in which the learner can choose several points for which the loss value will be observed. We remark that unlike previous work on multi-point bandit Agarwal et al. (2010); Duchi et al. (2015); Shamir (2017) we consider noisy feedback, and do not require additional assumptions on the loss function, as we show next.

The idea is to combine the method in Algorithm 1, with gradient estimation techniques for the bandit setting, by Flaxman et al. (2005). The approach of Flaxman et al. (2005) is based on constructing a simple estimate of the gradient, computed by evaluating the loss ℓ_t at a random point. Therefore, we obtain a smoothed approximation of the loss function. Note that since we construct a smoothed approximation of the loss, the smoothness assumption (Assumption 1) becomes redundant, as well the stochastic gradient oracle (Assumption 2). The following lemmas introduce the smoothed loss function and its properties:

Lemma 9 (Flaxman et al. (2005), Lemma 2.1) *Let $\mathcal{L}$ be a set of convex loss functions $\ell : \mathcal{Y} \to \mathbb{R}$ that are L-Lipschitz. For any $\ell \in \mathcal{L}$, define the function $\hat{\ell} \in \hat{\mathcal{L}}$ as follows: $\hat{\ell}(y) \triangleq \mathbb{E}_v[\ell(y + \delta v)]$, where v is a unit vector drawn uniformly at random, and $\delta > 0$. Then, $\hat{\ell}$ is differentiable, and:*

$$\nabla \hat{\ell}(y) = \mathbb{E}_v \left[\frac{d}{\delta} \ell(y + \delta v) v \right].$$

3. Moreover, projections according to distances other than the Euclidean distance can be defined, in particular w.r.t. Bregman divergences, and an analogue of the generalized Pythagorean theorem remains valid (see e.g., Lemma 11.3 *(3. Continued)* in Cesa-Bianchi and Lugosi (2006)). Thus, any class of loss functions that are measuring distance to some $y_t \in \mathcal{Y}$ based on a Bregman divergences, denote $\ell(y) = B_{\mathcal{R}}(y, y_t)$, corresponds to a suitable projection operation, that is simply $\Pi_{\mathcal{Y}}(z) \triangleq \arg\min_{y \in \mathcal{Y}} B_{\mathcal{R}}(y, z)$.

Lemma 10 *Let $\hat{\ell} \in \hat{\mathcal{L}}$, be a smoothed function as defined in Lemma 9. Then, the following holds:*

1. *$\hat{\ell}$ is convex, L-Lipschitz, and for any $y \in \mathcal{Y}$, $|\hat{\ell}(y) - \ell(y)| \leq \delta L$.*
2. *For any $y, y' \in \mathcal{Y}$, $\|\nabla\hat{\ell}(y) - \nabla\hat{\ell}(y')\| \leq \frac{d}{\delta} L\|y - y'\|$. Thus, $\hat{\ell}$ is $\frac{dL}{\delta}$-smooth.*
3. *For any $y \in \mathcal{Y}$, unit vector v, $\|\frac{d}{\delta}\ell(y + \delta v)v\| \leq \frac{dM}{\delta} \triangleq \sigma$.*

3.3. Algorithm and Analysis

The boosting algorithm maintains oracle access to N copies of a weak learning algorithm (see Definition 8), and iteratively produces predictions y_t, upon receiving an example x_t, by running a subroutine of a N-step optimization procedure. It generates a randomized gradient estimator $\mathbf{g}_{t,i}$ of function $\hat{\ell}_t(\cdot)$, a smoothed approximation of the loss function $\ell_t(\cdot)$,[4] as shown in Lemma 9, and Lemma 10. The estimator $\mathbf{g}_{t,i}$ is used in place of exact optimization with true gradients.

To update parameters, the gradient estimates are fed to the N weak learners as linear loss functions. Recall that $\mathcal{A}_i(\cdot)$ is not a function but rather the output of the algorithm $\mathcal{A}_i$ computed on the given example using its internal state, after having observed $\mathbf{g}_{1,i}...\mathbf{g}_{t-1,i}$. Intuitively, boosting guides each weak learner $\mathcal{A}_i$ to correct for mistakes of the preceding learner $\mathcal{A}_{i-1}$. The output prediction of the boosting algorithm (Line 13) relies on the projection operation, described in Assumption 3. A formal description is provided in Algorithm 2.

Algorithm 2 Online Gradient Boosting with Noisy Bandit Feedback

1: Maintain N weak learners $\mathcal{A}_1$,...,$\mathcal{A}_N$ (Definition 8).
2: Input: $\delta > 0$. Set step length $\eta_i = \frac{2}{i+1}$ for $i \in [N]$.
3: **for** $t = 1, \ldots, T$ **do**
4: Receive example x_t.
5: Define $y_t^0 = \mathbf{0}$.
6: **for** $i = 1$ to N **do**
7: Define $y_t^i = (1 - \eta_i)y_t^{i-1} + \eta_i \frac{1}{\gamma}\mathcal{A}_i(x_t)$.
8: Draw a unit vector v_t^i uniformly at random.
9: Receive bandit feedback: $\tilde{\ell}_t(y_t^{i-1} + \delta v_t^i)$.
10: Set $\mathbf{g}_{t,i} = \frac{d}{\delta}\tilde{\ell}_t(y_t^{i-1} + \delta v_t^i)v_t^i$.
11: Define linear loss function $\ell_t^i(y) = \mathbf{g}_{t,i}^\top \cdot y$, and pass $(x_t, \ell_t^i(\cdot))$ to weak learner $\mathcal{A}_i$.
12: **end for**
13: Output prediction $y_t := \Pi_{\mathcal{Y}}(y_t^N)$.
14: Receive bandit feedback $\tilde{\ell}_t(y_t)$.
15: **end for**

The following Theorem states the regret guarantees of Algorithm 2. We remark that although it uses expected regret as the performance metric, it can be converted to a guarantee that holds with high probability, with techniques similar to those used to obtain Theorem 3.

4. We assume that one can indeed query $\ell_t(\cdot)$ at any point $y + \delta v$. It is w.l.o.g. since a standard technique (see Agarwal et al. (2010); Hazan (2016)) is to simply run the learners $\mathcal{A}_i$ on a slightly smaller set $(1 - \xi)\mathcal{Y}$, where $\xi > 0$ is sufficiently large so that $y + \delta v$ must be in $\mathcal{Y}$. Since δ can be arbitrarily small, the additional regret/error incurred is arbitrarily small.

Theorem 11 *Given that the setting in 3.1, and assumption 3 hold, and given oracle access to N copies of an online weak learning algorithms (Definition 8) w.r.t. reference class $\mathcal{F}$ for linear losses, with $R_{\mathcal{A}}(T,\sigma)$ regret, then Algorithm 2 is an online learning algorithm w.r.t. reference class $CH(\mathcal{F})$ for convex losses ℓ_t, such that for any $f \in CH(\mathcal{F})$,*

$$\mathbb{E}[R_{\mathcal{B}}(T)] = \mathbb{E}\left[\sum_{t=1}^{T}\ell_t(\mathbf{y}_t) - \sum_{t=1}^{T}\ell_t(f(\mathbf{x}_t))\right] \leq \frac{2dLD^2T}{\delta\gamma^2 N} + \frac{R_{\mathcal{A}}(T, dM/\delta)}{\gamma} + 2T\delta L.$$

On the implications of Theorem 11. On the face of it, Theorem 11 converts a low regret algorithm into an algorithm with worse regret, at a computational cost of $O(N)$ per iteration. However, the main strength of this method is that it converts algorithms for a restricted setting into a significantly more general setting. In particular:

1. The input weak learners are γ-weak, guaranteeing a multiplicative loss guarantee. The resulting method is 1-competitive.

2. The input weak learners apply to linear loss functions. The resulting method applies to any convex loss sequence.

3. The input weak learners expect full information. The resulting method applies to the noisy multi-point bandit setting.

By setting the weak learning algorithm $\mathcal{A}$ to be any online learner for linear losses with a regret bound of $R_{\mathcal{A}}(T, dM/\delta) = O(\sqrt{T}dM/\delta)$, and by plugging in $\delta = T^{-1/4}$ and $N = \sqrt{T}$, an overall expected regret bound of $\mathbb{E}[R_{\mathcal{B}}(T)] = O(T^{3/4})$ is attained. Observe that the average regret $R_{\mathcal{B}}(T)/T$ converges to 0 as $T \to \infty$. While the requirement that $N \to \infty$ may raise concerns about computational efficiency, this is in fact analogous to the guarantee in the batch setting: the algorithms converge only when the number of boosting stages goes to infinity. Moreover, previous work on online boosting in the full information setting, gives a lower bound (Beygelzimer et al. (2015a), Theorem 4) which shows that $N \mapsto \infty$ is indeed necessary for sublinear regret.

Lastly, high probability bounds can also be obtained. Using a similar technique as in the OCO setting (Section 2, Theorem 3), a regret bound of $R_{\mathcal{B}}(T) = O(T^{3/4})$ which holds with high probability can be achieved in the boosting setting.

4. Discussion and future work

In this work, we have proposed a general framework for boosting regret minimization, when only limited information is provided. We demonstrated 2 implications: (a) a projection-free OCO algorithm with stochastic gradients, and (b) the first online boosting algorithm for regression problems in the multi-point noisy bandit setting. In this section, we discuss possible extensions of these results, that are left for future work.

Reinforcement learning. Consider the task of Reinforcement Learning (RL) in an adversarial Markov Decision Process (MDP) model (Even-Dar et al., 2009), which deals with online decision making against an adversary. These adversarial MDP models typically assume that the losses change arbitrarily over time, and that the transition function is unknown or adversarial (Rosenberg and Mansour, 2019a; Yadkori et al., 2013), and often also assume bandit feedback (Jin and Luo,

2019; Rosenberg and Mansour, 2019b). Therefore, this framework introduces an interesting application to the online boosting algorithm (Algorithm 2), when applied to the episodic-RL setting, i.e., where each episode $t \in [T]$ is treated as a separate time step.

Submodular optimization. Our framework considers the task of online learning for convex loss functions. A natural question is whether such an approach could be extended to the non-convex setting. Recently, continuous DR-submodular (diminishing returns) functions have been proposed as a broad class of non-convex functions which admit efficient approximate maximization routines, albeit exact maximization being NP-Hard (Bian et al., 2017). This setting capture many real-life applications, as well as a continuous relaxation of discrete submodular functions. A key property such functions hold is that they are concave in positive directions; thus they are amenable to efficient maximization via our framework, under similar assumptions. As previous work (Chen et al., 2018; Mokhtari et al., 2018) has demonstrated the tight connection between convexity and continuous DR-submodularity in the context of projection-free OCO, this suggests a natural extension of our methods to the submodular optimization setting.

References

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

Naman Agarwal, Nataly Brukhim, Elad Hazan, and Zhou Lu. Boosting for dynamical systems. *arXiv preprint arXiv:1906.08720*, 2019.

Alina Beygelzimer, Elad Hazan, Satyen Kale, and Haipeng Luo. Online gradient boosting. In *Advances in neural information processing systems*, pages 2458–2466, 2015a.

Alina Beygelzimer, Satyen Kale, and Haipeng Luo. Optimal and adaptive algorithms for online boosting. In *International Conference on Machine Learning*, pages 2323–2331, 2015b.

Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120, 2017.

Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 203–208, 1999.

Nataly Brukhim, Xinyi Chen, Elad Hazan, and Shay Moran. Online agnostic boosting via regret minimization. *arXiv preprint arXiv:2003.01150*, 2020.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823, 2018.

Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2047–2056, 2019.

Shang-Tse Chen, Hsuan-Tien Lin, and Chi-Jen Lu. An online boosting algorithm with theoretical justifications, 2012.

Shang-Tse Chen, Hsuan-Tien Lin, and Chi-Jen Lu. Boosting with online binary learners for the multiclass bandit problem. In *International Conference on Machine Learning*, pages 342–350, 2014.

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

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

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

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Dan Garber. Efficient online linear optimization with approximation algorithms. In *Advances in Neural Information Processing Systems*, pages 627–635, 2017.

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

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pages 5841–5851, 2017.

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

Elad Hazan and Satyen Kale. Projection-free online learning. In *29th International Conference on Machine Learning, ICML 2012*, pages 521–528, 2012.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. *COLT*, 2020.

Elad Hazan, Wei Hu, Yuanzhi Li, and Zhiyuan Li. Online improper learning with an approximation oracle. In *Advances in Neural Information Processing Systems*, pages 5652–5660, 2018.

Tiancheng Jin and Haipeng Luo. Learning adversarial mdps with bandit feedback and unknown transition. *arXiv preprint arXiv:1912.01192*, 2019.

Young Hun Jung and Ambuj Tewari. Online boosting algorithms for multi-label ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 279–287, 2018.

Young Hun Jung, Jack Goetz, and Ambuj Tewari. Online multiclass boosting. In *Advances in neural information processing systems*, pages 919–928, 2017.

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

Christian Leistner, Amir Saffari, Peter M Roth, and Horst Bischof. On robustness of on-line boosting-a competitive study. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1362–1369. IEEE, 2009.

Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.

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

Gergely Neu and Gábor Bartók. Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits. *The Journal of Machine Learning Research*, 17(1): 5355–5375, 2016.

Aviv Rosenberg and Yishay Mansour. Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*, pages 5478–5486, 2019a.

Aviv Rosenberg and Yishay Mansour. Online stochastic shortest path with bandit feedback and unknown transition function. In *Advances in Neural Information Processing Systems*, pages 2209–2218, 2019b.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. Cambridge university press, 2012. ISBN 9780262017183. doi: 10.1017/CBO9781107415324.004.

Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

Chu Wang, Yingfei Wang, Robert Schapire, et al. Functional frank-wolfe boosting for general loss functions. *arXiv preprint arXiv:1510.02558*, 2015.

Jiahao Xie, Zebang Shen, Chao Zhang, Hui Qian, and Boyu Wang. Stochastic recursive gradient-based methods for projection-free online learning. *arXiv preprint arXiv:1910.09396*, 2019.

Yasin Abbasi Yadkori, Peter L Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvári. Online learning in markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems*, pages 2508–2516, 2013.

Daniel T Zhang, Young Hun Jung, and Ambuj Tewari. Online multiclass boosting with bandit feedback. *arXiv preprint arXiv:1810.05290*, 2018.

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

Appendix A. Technical Lemmas

In this section we prove several useful claims and lemmas that are used in the main analysis.

A.1. Proof of Lemma 6

Proof We have,

$$\begin{aligned}\ell(z^i) &= \ell(z^{i-1} + \eta_i(\frac{1}{\gamma}z_i - z^{i-1})) \\ &\leq \ell(z^{i-1}) + \eta_i \nabla\ell(z^{i-1})^\top \cdot (\frac{1}{\gamma}z_i - z^{i-1}) + \frac{\eta_i^2\beta}{2}\|\frac{1}{\gamma}z_i - z^{i-1}\|^2 \\ &\leq \ell(z^{i-1}) + \eta_i \nabla\ell(z^{i-1})^\top \cdot (\frac{1}{\gamma}z_i - z^{i-1}) + \frac{\eta_i^2\beta D^2}{2\gamma^2},\end{aligned} \tag{5}$$

where the inequalities follow from the β-smoothness of ℓ, and the bound on the set $\mathcal{Z}$, respectively. Observe that,

$$\begin{aligned}\nabla\ell(z^{i-1})^\top(\frac{1}{\gamma}z_i - z^{i-1}) &= g_i^\top(\frac{1}{\gamma}z_i - z^{i-1}) + (\nabla\ell(z^{i-1}) - g_i)^\top(\frac{1}{\gamma}z_i - z^{i-1}) \\ &\quad \text{(by adding and subtracting the term: } g_i^\top(\frac{1}{\gamma}z_i - z^{i-1})) \\ &= g_i^\top(\frac{1}{\gamma}z_i - z) + g_i^\top(z - z^{i-1}) + (\nabla\ell(z^{i-1}) - g_i)^\top(\frac{1}{\gamma}z_i - z^{i-1}) \\ &\quad \text{(by adding and subtracting the term: } g_i^\top z) \\ &= g_i^\top(\frac{1}{\gamma}z_i - z) + \nabla\ell(z^{i-1})^\top(z - z^{i-1}) + (\nabla\ell(z^{i-1}) - g_i)^\top(\frac{1}{\gamma}z_i - z) \\ &\quad \text{(by adding and subtracting the term: } \nabla\ell(z^{i-1})^\top z) \\ &\leq g_i^\top(\frac{1}{\gamma}z_i - z) + \ell(z) - \ell(z^{i-1}) + (\nabla\ell(z^{i-1}) - g_i)^\top(\frac{1}{\gamma}z_i - z) \\ &\quad \text{(by convexity, } \nabla\ell(z^{i-1})^\top \cdot (z - z^{i-1}) \leq \ell(z) - \ell(z^{i-1})).\end{aligned} \tag{6}$$

Combining (5) and (6), and the definition of ζ_i we have that,

$$\Big(\ell(z^i) - \ell(z)\Big) \leq (1 - \eta_i)\Big(\ell(z^{i-1}) - \ell(z)\Big) + \frac{\eta_i^2\beta D^2}{2\gamma^2} + \eta_i\Big(g_i^\top(\frac{1}{\gamma}z_i - z) + \zeta_i\Big).$$

■

A.2. Proof of Claim 7

Proof We prove by induction over $i > 0$. For $i = 1$, since $\eta_1 = 1$, the assumption implies that $\phi_1 \leq \frac{C_1}{2} + C_2$. Thus, the base case of the induction holds true. Now assume the claim holds for

$i = k$, and we will prove it holds for $i = k + 1$. By the induction step,

$$\begin{aligned}\phi_{k+1} &\leq \Big(1 - \frac{2}{k+2}\Big)\phi_k + \frac{2C_1}{(k+2)^2} + \frac{2C_2}{k+2} \\ &\leq \frac{k}{k+2}\Big(\frac{2C_1}{k+1} + C_2\Big) + \frac{2C_1}{(k+2)^2} + \frac{2C_2}{k+2} \\ &= \frac{2C_1}{k+2}\Big(\frac{k}{k+1} + \frac{1}{k+2}\Big) + C_2 \leq \frac{2C_1}{k+2} + C_2.\end{aligned}$$

■

Appendix B. High probability bounds for Projection-Free OCO with Stochastic Gradients

In this section we give a high-probability regret bound to Algorithm 1. Observe that when the variance of the base OLO algorithm is unbounded, the regret guarantees cannot hold with high probability. Thus, we slightly modify the OLO definition to hold w.h.p. This is w.l.o.g as there are projection-free OLO algorithm for which such guarantees hold, as we describe in Theorem 3.

Definition 12 *Let $\mathcal{L}'$ denote a class of linear loss functions, $\ell' : \mathcal{K} \to \mathbb{R}$. An online learning algorithm $\mathcal{A}$ is an **Online Linear Optimizer (OLO)** for $\mathcal{K}$ w.r.t. $\mathcal{L}'$, if for any $\rho \in (0,1)$, and any sequence of losses $\ell'_1, ..., \ell'_T \in \mathcal{L}'$, w.p. at least $1-\rho$, the algorithm has regret w.r.t. $\mathcal{L}'$, $R_{\mathcal{A}}(T)$ that is sublinear in T.*

We can now derive the following proposition (corresponding to Proposition 5 of the expected case):

Proposition 13 *Given that assumptions 1 - 2 hold, and given oracle access to N copies of an OLO algorithm for linear losses, with $R_{\mathcal{A}}(T)$ regret, Algorithm 1 is an OCO algorithm which only requires $N = O(\sqrt{T})$ stochastic gradient oracle calls per iteration, such that for any $\rho \in (0,1)$, and any sequence of convex losses ℓ_t over convex set $\mathcal{K}$, w.p. at least $1-\rho$,*

$$\sum_{t=1}^{T} \ell_t(x_t) - \inf_{x^* \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(x^*) \leq \frac{2\beta D^2 T}{N} + R_{\mathcal{A}}(T) + (\sigma + G)D\sqrt{2T\log(4N/\rho)}.$$

Proof Let $x_{t,i} \in \mathcal{K}$ be the output of the OLO algorithm $\mathcal{A}_i$ at time t, and let x^* be any point in $\mathcal{K}$. The regret definition of $\mathcal{A}_i$ (Definition 12), and the definition of $\ell^i_t(\cdot)$ in Algorithm 1, imply that for $\rho \in (0,1)$ we have that, w.p. at least $1 - \rho/(2N)$,

$$\sum_{t=1}^{T} \mathbf{g}_{t,i}^\top \cdot x_{t,i} \; - \sum_{t=1}^{T} \mathbf{g}_{t,i}^\top \cdot x^* \; \leq \; R_{\mathcal{A}}(T). \tag{7}$$

By applying Lemma 6, and by the OLO guarantee (7), we get that,

$$\Delta_i \;\; \leq \;\; (1-\eta_i)\Delta_{i-1} + \frac{\eta_i^2 \beta D^2}{2} T + \eta_i \Big(R_{\mathcal{A}}(T) + \sum_{t=1}^{T} \boldsymbol{\zeta}_{t,i} \Big). \tag{8}$$

where $\Delta_i \triangleq \sum_{t=1}^{T} \ell_t(\mathbf{x}_t^i) - \ell_t(x^*)$, and $\boldsymbol{\zeta}_{t,i} \triangleq (\nabla \ell_t(\mathbf{x}_t^{i-1}) - \mathbf{g}_{t,i})^\top \cdot (x_{t,i} - x^*)$, for $i \in [N]$. By applying the union bound, the above inequality holds for all $i \in [N]$, with probability at least $1-\rho/2$.

For any fixed $i \in [N]$, Observe that $\mathbb{E}\left[\boldsymbol{\zeta}_{t,i} | \mathcal{I}_{t-1}^i\right] = \mathbb{E}\left[(\nabla \ell_t(\mathbf{x}_t^{i-1}) - \mathbf{g}_{t,i})^\top \cdot (x_{t,i} - x^*) | \mathcal{I}_{t-1}^i\right] = 0$ by Lemma 4. Therefore, $\{\boldsymbol{\zeta}_{t,i}\}_{t=1}^T$ is a martingale difference sequence. Moreover, by the Cauchy-Schwartz inequality, we have,

$$|\boldsymbol{\zeta}_{t,i}| \leq \|\nabla \ell_t(\mathbf{x}_t^{i-1}) - \mathbf{g}_{t,i}\| \cdot \|x_{t,i} - x^*\| \leq \Big(\|\nabla \ell_t(\mathbf{x}_t^{i-1})\| + \|\mathbf{g}_{t,i}\|\Big) \cdot \|x_{t,i} - x^*\| \leq (G+\sigma) \cdot D = c_t,$$

where the second inequality follows from the triangle inequality, and the last inequality follows from the diameter bound D on the set $\mathcal{K}$, the bound on the gradient norm G, and the bound on the stochastic gradient estimate (Assumption 2). Let $\lambda = (\sigma + G)D\sqrt{2T\log(4N/\rho)}$, by the Azuma-Hoeffding inequality,

$$\mathbb{P}\left[\left|\sum_{t=1}^{T} \boldsymbol{\zeta}_{t,i}\right| \geq \lambda\right] \leq 2\exp\left(-\frac{\lambda^2}{2\sum_{t=1}^{T} c_t^2}\right) = \rho/2N.$$

Observe that, by applying the union bound, the above inequality holds for all $i \in [N]$, with probability at least $1 - \rho/2$. Therefore, by combining the above with (8), applying union bound, we get that w.p. at least $1 - \rho$, we have for all $i \in [N]$,

$$\Delta_i \;\; \leq \;\; (1 - \eta_i)\Delta_{i-1} + \frac{\eta_i^2 \beta D^2}{2} T + \eta_i \Big(R_{\mathcal{A}}(T) + (\sigma + G)D\sqrt{2T\log(4N/\rho)}\Big).$$

Applying Claim 7, and setting $i = N$ yields that,

$$\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^*) \leq \frac{2\beta D^2 T}{N} + R_{\mathcal{A}}(T) + (\sigma + G)D\sqrt{2T\log(4N/\rho)}. \tag{9}$$

■

B.1. Proof of Theorem 3

Proof The proof of Theorem 3 is a direct Corollary of Proposition 13, by plugging *Follow the Perturbed Leader* Kalai and Vempala (2005) with high probability guarantees (e.g., Neu and Bartók (2016)) as the OLO algorithm required for Algorithm 1. We get that the regret of the base algorithms $\mathcal{A}_i$ is $R_{\mathcal{A}}(T) = O(\sigma D\sqrt{T})$, where D is the diameter of the set $\mathcal{K}$, and σ is the bound on the stochastic gradient norm (Assumption 2). Thus, by setting $N = \frac{\beta D}{\sigma}\sqrt{T}$, we get that w.p. at least $1 - \rho$,

$$\begin{aligned}\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^*) &\leq 2\sigma D\sqrt{T} + O(\sigma D\sqrt{T}) + (\sigma + G)D\sqrt{2T\log\left(\beta DT/(\sigma\rho)\right)} \\ &= O\Big(\sigma D\sqrt{T\log\left(\beta DT/(\sigma\rho)\right)}\Big).\end{aligned}$$

■

B.2. Proof of Lemma 10

Proof Below are the proofs of each item:

1. The fact that $\hat{\ell}$ is convex, L-Lipschitz is immediate from its definition and the assumptions on ℓ. The inequality follows from v being a unit vector and that ℓ is assumed to be L-Lipschitz.

2. For any $x, x' \in \mathbb{R}^d$,

$$\begin{aligned}\|\nabla\hat{\ell}(x) - \nabla\hat{\ell}(x')\| &= \frac{d}{\delta}\|\mathbb{E}[(\ell(x+\delta v) - \ell(x'+\delta v))v]\| \\ &\leq \frac{d}{\delta}\mathbb{E}\Big[\|(\ell(x+\delta v) - \ell(x'+\delta v))v\|\Big] \\ &\leq \frac{d}{\delta}\mathbb{E}\Big[|\ell(x+\delta v) - \ell(x'+\delta v)|\Big] \\ &\leq \frac{d}{\delta}L\|x - x'\|,\end{aligned}$$

   where the first inequality follows from Jensen's Inequality, the second inequality follows from the fact that v is a unit vector, and the next inequality from ℓ being L-Lipschitz. This property implies that the function $\hat{\ell}$ is $\frac{dL}{\delta}$-smooth.

3. For any $x \in \mathbb{R}^d$, and unit vector u, $\|\nabla\hat{\ell}(x) - \big(\frac{d}{\delta}\tilde{\ell}(x+\delta u)u\big)\| \leq \|\frac{d}{\delta}\tilde{\ell}(x+\delta u)u\| + \|\nabla\hat{\ell}(x)\|$. Note that by the fact that $\hat{\ell}$ is L-Lipschitz, we have $\|\nabla\hat{\ell}(x)\| \leq L$. The first term can be bounded as follows:

$$\begin{aligned}\|\frac{d}{\delta}\tilde{\ell}(x+\delta u)u\| &= \frac{d}{\delta}\tilde{\ell}(x+\delta u)\|u\| \leq \frac{d}{\delta}\tilde{\ell}(x+\delta u) \\ &= \frac{d}{\delta}\big(\ell(x+\delta u) + w\big) \leq \frac{d}{\delta}2M,\end{aligned}$$

   where the first inequality follows from the fact that u is a unit vector, the equality follows from the definition of $\tilde{\ell}$, and the last inequality follows from the bounds on ℓ and $w \sim \mathcal{D}$. Therefore, we have,

$$\|\nabla\hat{\ell}(x) - \big(\frac{d}{\delta}\tilde{\ell}(x+\delta u)u\big)\| \leq \frac{2dM}{\delta} + L.$$

■

Theorem 14 *Algorithm 1 is a projection-free OCO algorithm for the bandit setting, with $N = \sqrt{T}$ bandit feedback values per round, such that for any $\rho \in (0,1)$, and any sequence of convex losses $\ell_t \in \mathcal{L}$ over convex set $\mathcal{K}$, w.p. at least $1-\rho$,*

$$\sum_{t=1}^{T}\ell_t(x_t) - \inf_{x^*\in\mathcal{K}}\sum_{t=1}^{T}\ell_t(x^*) \leq O\left(dMLD^2T^{3/4}\sqrt{\log(T/\rho)}\right) = \tilde{O}(T^{3/4}).$$

B.3. Proof of Theorem 14

Proof Observe that by Lemma 9, we have that Assumptions 1-2 are redundant, and so Lemma 4 and Proposition 13 hold for losses $\hat{\ell}_t \in \hat{\mathcal{L}}$, with $G = L$, $\sigma = dM/\delta$, and $\beta = dL/\delta$, by Lemma 10. Thus, we have that w.p at least $1-\rho$,

$$\sum_{t=1}^{T} \hat{\ell}_t(x_t) - \hat{\ell}_t(x^*) \leq \frac{2dLD^2T}{\delta N} + R_{\mathcal{A}}(T) + (dM/\delta + L)D\sqrt{2T\log(4N/\rho)}. \tag{10}$$

Now, observe that,

$$\begin{aligned}
\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^*) &\leq \sum_{t=1}^{T} \hat{\ell}_t(x_t) - \hat{\ell}_t(x^*) + 2T\delta L && \text{(By Lemma 10 (1))} \\
&\leq \frac{2dLD^2T}{\delta N} + R_{\mathcal{A}}(T) + (dM/\delta + L)D\sqrt{2T\log(4N/\rho)} + 2T\delta L && \text{(By (10))} \\
&\leq \frac{2dLD^2T}{\delta N} + O(dMD\sqrt{T}/\delta) + (dM/\delta + L)D\sqrt{2T\log(4N/\rho)} + 2T\delta L && (11)
\end{aligned}$$

where the last inequality follows by plugging *Follow the Perturbed Leader* Kalai and Vempala (2005) with high probability guarantees (e.g., Neu and Bartók (2016)) as the OLO algorithm required for Algorithm 1. Thus, the base algorithms $\mathcal{A}_i$'s regret is $R_{\mathcal{A}}(T) = O(dMD\sqrt{T}/\delta)$.

Lastly the results follows by plugging in $\delta = T^{-1/4}$ and $N = \sqrt{T}$ into Equation (11), to obtain regret of at most $O\left(dMLD^2T^{3/4}\sqrt{\log(T/\rho)}\right) = \tilde{O}(T^{3/4})$, w.p at least $1-\rho$. ■

Appendix C. Online Boosting: Proofs

In this section we give the full analysis of the Algorithm and results given in Section 3. For simplicity assume an oblivious adversary (can also be shown to hold for an adaptive one). Let $(x_1, \ell_1), ..., (x_T, \ell_T)$ be **any** sequence of examples and losses. Observe that the only sources of randomness at play are: the weak learners' ($\mathcal{A}_i$'s) internal randomness, the random unit vectors v_t^i, and the additive zero-mean noise for any bandit feedback. The analysis below is given in expectation with respect to all these random variables.

Lemma 15 *For any $t \in [T]$ and $i \in [N]$, let $\mathbf{g}_{t,i}$ be the stochastic gradient estimate used in Algorithm 1, s.t. $\mathbb{E}[\mathbf{g}_{t,i}] = \nabla\hat{\ell}(y_t^{i-1})$, and $\ell_t^i(\mathbf{y}) = \mathbf{g}_{t,i}^\top \cdot \mathbf{y}$. Then, we have,*

$$\mathbb{E}\left[\ell_t^i\big(\mathcal{A}_i(x_t)\big)\right] = \mathbb{E}\left[\nabla\hat{\ell}(y_t^{i-1})^\top \cdot \mathcal{A}_i(x_t)\right].$$

Proof Let $\mathcal{I}_t^{i-1}$ denotes the σ-algebra measuring all sources of randomness up to time t and learner $i-1$; i.e., the internal randomness of weak learners $\mathcal{A}_1, ..., \mathcal{A}_{i-1}$, the the random unit vectors $v_1^j, ..., u_t^j$, for all $j < 1$, and the noise terms $w_{1,j}, ..., w_{t,j}$ for all $j < 1$. Then,

$$\begin{aligned}
\mathbb{E}\big[\ell_t^i(\mathcal{A}_i(x_t))\big] &= \mathbb{E}\Big[\mathbf{g}_{t,i}^\top \cdot \mathcal{A}_i(x_t)\Big] && \text{(definition of } \ell_t^i(\cdot)) \\
&= \mathbb{E}\Big[\Big(\frac{d}{\delta}\tilde{\ell}_t(y_t^{i-1} + \delta v_t^i) \cdot v_t^i\Big)^\top \cdot \mathcal{A}_i(x_t)\Big] && \text{(definition of } \mathbf{g}_{t,i}) \\
&= \mathbb{E}\Big[\Big(\frac{d}{\delta}\ell_t(y_t^{i-1} + \delta v_t^i) \cdot v_t^i\Big)^\top \cdot \mathcal{A}_i(x_t)\Big] \\
& && \text{(since } \tilde{\ell}(z) = \ell(z) + w, \text{ with } w \text{ i.i.d., } \mathbb{E}[w] = 0) \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\left[\mathbb{E}\Big[\Big(\frac{d}{\delta}\ell_t(y_t^{i-1} + \delta v_t^i) \cdot v_t^i\Big)^\top \cdot \mathcal{A}_i(x_t)\Big|\mathcal{I}_t^{i-1}\Big]\right] \\
& && \text{(by law of total expectation)} \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\left[\mathbb{E}_{v_t^i}\Big[\frac{d}{\delta}\ell_t(y_t^{i-1} + \delta v_t^i) \cdot v_t^i\Big|\mathcal{I}_t^{i-1}\Big]^\top \cdot \mathbb{E}\Big[\mathcal{A}_i(x_t)\Big|\mathcal{I}_t^{i-1}\Big]\right] \\
& && \text{(by conditional independence)} \\
&= \mathop{\mathbb{E}}_{\mathcal{I}_t^{i-1}}\left[\nabla\hat{\ell}_t(y_t^{i-1})^\top \cdot \mathbb{E}\Big[\mathcal{A}_i(x_t)\Big|\mathcal{I}_t^{i-1}\Big]\right] && \text{(by Lemma 9)} \\
&= \mathbb{E}\Big[\nabla\hat{\ell}_t(y_t^{i-1})^\top \cdot \mathcal{A}_i(x_t)\Big]
\end{aligned}$$

■

C.1. Proof of Theorem 11

Proof First, note that for any $i = 1, 2, \ldots, N$, since ℓ_t^i is a linear function, we have

$$\inf_{f \in \mathrm{CH}(\mathcal{F})} \sum_{t=1}^T \ell_t^i(f(\mathbf{x}_t)) \;=\; \inf_{f \in \mathcal{F}} \sum_{t=1}^T \ell_t^i(f(\mathbf{x}_t)).$$

Let f be any function in $\mathrm{CH}(\mathcal{F})$. The equality above, the regret bound of the weak learner $\mathcal{A}^i$ for $\mathcal{F}$ (see Definition 8), and the definition of $\ell_t^i(\cdot)$ in Algorithm 2, imply that:

$$\mathbb{E}\left[\sum_{t=1}^T \mathbf{g}_{t,i}^\top \cdot \mathcal{A}^i(\mathbf{x}_t) \;- \gamma \sum_{t=1}^T \mathbf{g}_{t,i}^\top \cdot f(\mathbf{x}_t)\right] \;\leq\; R_{\mathcal{A}}(T). \tag{12}$$

Now define, for $i = 0, 1, 2, \ldots, N$, $\Delta_i = \sum_{t=1}^T \hat{\ell}_t(\mathbf{y}_t^i) - \hat{\ell}_t(f(\mathbf{x}_t))$. By applying Lemma 6, we get,

$$\Delta_i \;\leq\; (1 - \eta_i)\Delta_{i-1} + \frac{\eta_i^2 \beta D^2}{2\gamma^2} T + \eta_i \sum_{t=1}^T \Big(\mathbf{g}_{t,i}^\top (\frac{1}{\gamma}\mathcal{A}^i(\mathbf{x}_t) - f(\mathbf{x}_t)) + \boldsymbol{\zeta}_{t,i}\Big)$$

Dataset	Full Information		Bandit		Relative Decrease
	Baseline (OGD)	Online Boosting	Baseline (N-FKM)	Online Boosting	
abalone	3.708 $\pm$.027	3.71 $\pm$.006	12.21 $\pm$.210	11.68 $\pm$.154	4.34%
adult	0.154 $\pm$.003	0.151 $\pm$.002	0.161 $\pm$.003	0.150 $\pm$.001	6.83%
census	0.160 $\pm$.002	0.032 $\pm$.001	0.163 $\pm$.001	0.105 $\pm$.020	35.6%
letter	0.507 $\pm$.008	0.498 $\pm$.002	0.522 $\pm$.006	0.517 $\pm$.003	0.95%
slice	0.042 $\pm$.0001	0.040 $\pm$.0001	0.049 $\pm$.001	0.045 $\pm$.001	8.16%

Table 2: Average loss of boosting and baseline algorithms on various datasets, with standard deviation. Relative loss decrease of boosting compared to baseline, shown for bandit setting.

where $\boldsymbol{\zeta}_{t,i} \triangleq (\nabla \ell_t(\mathbf{y}_t^{i-1}) - \mathbf{g}_{t,i})^\top \cdot (\mathcal{A}^i(\mathbf{x}_t) - f(\mathbf{x}_t))$. Take expectation on both sides. By Lemma 15, we have $\mathbb{E}[\boldsymbol{\zeta}_{t,i}] = 0$, and by the weak learning guarantee (12), we get that,

$$\mathbb{E}\big[\Delta_i\big] \;\leq\; (1-\eta_i)\mathbb{E}\big[\Delta_{i-1}\big] + \frac{\eta_i^2 \beta D^2}{2\gamma^2} T + \frac{\eta_i}{\gamma} R_{\mathcal{A}}(T)$$

By Claim 7 (with $\phi_i = \mathbb{E}[\Delta_i]$), we get,

$$\mathbb{E}\big[\Delta_i\big] \leq \frac{2\beta D^2 T}{\gamma^2 (i+1)} + \frac{R_{\mathcal{A}}(T)}{\gamma}. \tag{13}$$

Lastly, observe that,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\mathbf{y}_t) - \ell_t(f(\mathbf{x}_t))\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \hat{\ell}_t(\mathbf{y}_t) - \hat{\ell}_t(f(\mathbf{x}_t))\right] + 2T\delta L && \text{(by Lemma 10 (2))} \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \hat{\ell}_t(\mathbf{y}_t^N) - \hat{\ell}_t(f(\mathbf{x}_t))\right] + 2T\delta L && \text{(by Assumption 3)} \\
&\leq \frac{2\beta D^2 T}{\gamma^2 N} + \frac{R_{\mathcal{A}}(T)}{\gamma} + 2T\delta L && \text{(by (13), for } i = N) \\
&\leq \frac{2dLD^2 T}{\delta \gamma^2 N} + \frac{R_{\mathcal{A}}(T)}{\gamma} + 2T\delta L && \text{(by Lemma 10 (3))}
\end{aligned}$$

■

Appendix D. Experiments

While the focus of this paper is theoretical investigation of online boosting and projection-free algorithms with limited information, we have also performed experiments to evaluate our algorithms. We focused our empirical investigation on the more challenging task of Online Boosting with bandit feedback, proposed in Section 3. Algorithm 2 was implemented in NumPy, and the weak online

learner was a linear model updated with FKM Flaxman et al. (2005), online projected gradient descent with spherical gradient estimators. To facilitate a fair comparison to a baseline, we provided an FKM model with a N-point noisy bandit feedback, where N is the number of weak learners of the corresponding boosting method. We denote this baseline as N-FKM. We also compare against the full information setting, which amounts to the method used in previous work (Beygelzimer et al. (2015a), Algorithm 2), and compared to a linear model baseline updated with online gradient descent (OGD). The main strength of our results compared to previous work of online boosting Beygelzimer et al. (2015a) is the ability to handle partial information without a large loss of accuracy, as demonstrated in Table 2. Table 2 summarizes the average squared loss and the standard deviation, and the last column refers to the relative loss decrease on average, of boosting in the bandit setting compared to the N-FKM baseline.

The experiments we carry out were proposed by Beygelzimer et al. (2015a) for evaluating online boosting. They are composed of several data sets for regression and classification tasks, obtained from the UCI machine learning repository (and further described in the supplementary material). We remark that our setting assumes *noisy* bandit feedback. Therefore, the true label cannot be easily recovered even for binary label classification, and the task is significantly harder than in the full information setting. This is also evident by the comparison over these datasets with previous work (where the online boosting algorithm in the full information setting performs better since it is given the true loss) and with the baseline in the bandit setting (which performs worse without boosting), as shown in Table 2 .

For each experiment, reported are average results over 20 different runs. In the bandit setting, each loss function evaluation was obtained with additive noise, uniform on $[\pm.1]$, and gradients were evaluated as in Algorithm 2. The only hyper-parameters tuned were the learning rate, N the number of weak learners, and the smoothing parameter δ. Our theoretical guarantees determine that only $N := \sqrt{T}$ iterations need to be used, and N is a pre-specified parameter. Empirically, we find that a small number of iterations is sufficient, much smaller than $\sqrt{T}$, and was set to at most 30, even for very large datasets ($T \approx 300K$). Parameters were tuned based on progressive validation loss on half of the dataset; reported is progressive validation loss on the remaining half. Progressive validation is a standard online validation technique, where each training example is used for testing before it is used for updating the model Blum et al. (1999).

D.1. Experimental setup description

The datasets were taken from the UCI machine learning repository, and their statistics are detailed below, along with the link to a downloadable version of each dataset.

Dataset	#Instances	#Features	Downloadable version	Task	Label range
abalone	4,177	10	Link	regression	$[1, 29]$
adult	48,842	105	Link	classification	$[0, 1]$
census	299,284	401	Link	classification	$[0, 1]$
letter	20,000	16	Link	classification	$[-1, 1]$
slice	53,500	385	Link	regression	$[0, 1]$

Algorithm 2 was implemented in NumPy, and the weak online learner was a linear model updated with FKM Flaxman et al. (2005), online projected gradient descent with spherical gradient estimators. To facilitate a fair comparison to a baseline, we provided an FKM model with a N-

point noisy bandit feedback, where N is the number of weak learners of the corresponding boosting method. We denote this baseline as N-FKM. We also compare against the full information setting, which amounts to the method used in previous work (Beygelzimer et al. (2015a), Algorithm 2), and compared to a linear model baseline updated with online gradient descent (OGD).

The experiments we carry out were proposed by Beygelzimer et al. (2015a) for evaluating online boosting, they are composed of several data sets for regression and classification tasks, obtained from the UCI machine learning repository. For each experiment, reported are average results over 20 different runs. In the bandit setting, each loss function evaluation was obtained with additive noise, uniform on $[\pm.1]$, and gradients were evaluated as in Algorithm 2. The only hyper-parameters tuned were the learning rate, N the number of weak learners, and the smoothing parameter δ:

- N was set in the range of $[5, 30]$.
- δ was set to $1/2$ in all the experiments.
- Learning rate at time t is lr $*t^{-c}$ where lr and c were set in the ranges [1e-04, 0.1], $[.25, 1]$.