
Boosting for Online Convex Optimization

Elad Hazan^{1,2} Karan Singh³

Abstract

We consider the decision-making framework of online convex optimization with a very large number of experts. This setting is ubiquitous in contextual and reinforcement learning problems, where the size of the policy class renders enumeration and search within the policy class infeasible. Instead, we consider generalizing the methodology of online boosting. We define a weak learning algorithm as a mechanism that guarantees multiplicatively approximate regret against a base class of experts. In this access model, we give an efficient boosting algorithm that guarantees near-optimal regret against the convex hull of the base class. We consider both full and partial (a.k.a. bandit) information feedback models. We also give an analogous efficient boosting algorithm for the i.i.d. statistical setting. Our results simultaneously generalize online boosting and gradient boosting guarantees to contextual learning model, online convex optimization and bandit linear optimization settings.

1. Introduction

In the classical problem of prediction from expert advice (Cesa-Bianchi et al., 1997), a learner iteratively makes decisions and receives loss according to an arbitrarily chosen loss function. Learning in terms of guaranteeing an absolute bound on the loss incurred would be a hopeless task, but the learner is assisted by a pool of experts (or hypotheses). The goal of the learner is thus to minimize regret, or the difference of total loss to that of the best expert in hindsight.

For this canonical problem, numerous methods have been devised and refined, some notably based on the multiplicative weights algorithm (Littlestone & Warmuth, 1994; Arora

et al., 2012). It is well established that the regret can be bounded by $O(\sqrt{T \log |\mathcal{H}|})$, where $|\mathcal{H}|$ is the number of experts, and that this is tight. However, in many problems of interest, the class of experts is too large to efficiently manipulate. This is particularly evident in contextual learning, where the experts are *policies* – functions mapping contexts to action. In such instances, even if a regret bound of $O(\sqrt{T \log |\mathcal{H}|})$ is meaningful, the algorithms achieving this bound are computationally inefficient; their running time is linear in $|\mathcal{H}|$.

One approach to deal with this computational difficulty is to grant the learner a best-in-hindsight (or ERM) oracle. This has been explored in both the stochastic bandit (Langford & Zhang, 2008; Agarwal et al., 2014; Dudik et al., 2011) and (transductive) online settings (Rakhlin & Sridharan, 2016; Syrgkanis et al., 2016b;a; Hazan & Koren, 2016).

In this paper, we consider a different approach towards addressing computational intractability, motivated by the observation that it is often possible to design simple *rules-of-thumb* (Viola & Jones, 2001) that perform slightly better than random guesses. We propose that the learner has access to a “weak learner” - an computationally cheap mechanism capable of guaranteeing multiplicatively approximate regret against a base hypotheses class. Such considerations arise in the theory of boosting (Schapire & Freund, 2012b), which was originally devised in the context of binary classification in the statistical/i.i.d. setting. A weak learner, defined as one which is capable of better-than-random classification, is used by a boosting algorithm to create an accurate classifier. By analogy, the regret-based notion of weak learner we define here is a natural extension of the concept of weak learning to online convex optimization.

We design efficient algorithms that when provided weak learners compete with the convex hull of the base hypotheses class with near-optimal regret. We further consider different information feedback models: both full gradient feedback, as well as linear bandit (function value) feedback, and derive efficient sublinear regret algorithms for these settings.

1.1. Setting and contributions

The setting of online convex optimization (OCO) generalizes the problem of prediction from expert advice to a general convex decision set $\mathcal{K} \subseteq \mathbb{R}^d$, and adversarially

^{*}Equal contribution ¹Google AI Princeton, Princeton, NJ, USA ²Princeton University, Princeton, NJ, USA ³Microsoft Research, Redmond, WA, USA. Correspondence to: Elad Hazan <ehazan@cs.princeton.edu>, Karan Singh <karansingh@microsoft.com>.

chosen convex loss functions $f_t : \mathbb{R}^d \mapsto \mathbb{R}$. We consider a hypothesis class $\mathcal{H} \subseteq \mathcal{C} \mapsto \mathcal{K}$, where each hypothesis $h \in \mathcal{H}$ maps a context (or feature vector) in the context set $c_t \in \mathcal{C}$ to an action $h(c_t) \in \mathcal{K}$.

Definition 1. An online learning algorithm \mathcal{W} is a γ -**weak OCO learner (WOCL)** for a class \mathcal{H} and edge $\gamma \in (0, 1)$, if for any sequence of contexts chosen from \mathcal{C} and unit¹ linear loss functions f_1, \dots, f_T , we have

$$\sum_{t=1}^T f_t(\mathcal{W}(c_t)) \leq \gamma \cdot \min_{h \in \mathcal{H}} \sum_{t=1}^T f_t(h(c_t)) + (1 - \gamma) \sum_{t=1}^T f_t^\mu + \text{Regret}_T(\mathcal{W}),$$

where $f^\mu = \int_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x}) d\mu$ is the average value of f under the uniform distribution μ .

In classical boosting, a weak learner is an algorithm that offers an edge over a random guess. Since f_t^μ is precisely the loss associated with a uniform random guess, the above definition is a generalization of the classic notion. In addition, note that such a definition is invariant under constant offsets to the function value.

It is convenient to henceforth assume that the convex decision set is centered, that is $\int_{\mathbf{x} \in \mathcal{K}} \mathbf{x} d\mu = 0$. If so, note that for any linear function $f : \mathcal{K} \rightarrow \mathcal{R}$, $f_t^\mu = f(\int_{\mathbf{x}} \mathbf{x} d\mu) = 0$. Under this assumption, we can rephrase γ -WOCL as

$$\sum_{t=1}^T f_t(\mathcal{W}(c_t)) \leq \gamma \cdot \min_{h \in \mathcal{H}} \sum_{t=1}^T f_t(h(c_t)) + \text{Regret}_T(\mathcal{W}). \quad (1)$$

Let $\text{CH}(\mathcal{H})$ be the convex hull of the hypothesis class. Our goal is to design an algorithm \mathcal{A} which minimizes regret vs. the best hypothesis in the convex hull of the base class:

$$\text{Regret} = \sum_{t=1}^T f_t(\mathcal{A}(c_t)) - \min_{h \in \text{CH}(\mathcal{H})} \sum_{t=1}^T f_t(h(c_t)).$$

In this model, our main result is an efficient algorithm, whose guarantee is given by:

Theorem 2 (Main). *Algorithm 1 with parameters γ, N , maintains N copies of γ -WOCL, and generates a sequence actions $\mathbf{x}_1, \dots, \mathbf{x}_T$, such that*

$$\text{Regret} = O\left(\frac{T}{\gamma\sqrt{N}} + \frac{1}{\gamma} \text{Regret}_T(\mathcal{W})\right).$$

¹ $f(x)$ is a unit loss function if $\max_{x \in \mathcal{K}} f(x) - \min_{x \in \mathcal{K}} f(x) \leq 1$. This scaling is w.l.o.g. and all results hold with a different constant otherwise.

The formal theorem which we state and prove below has explicit constants that depend on the diameter of \mathcal{K} (as opposed to $|\mathcal{H}|$) and the Lipschitz constants of the loss functions. These are hidden for brevity. Notably, Algorithm 1 achieves the following goals:

1. Its running time does **not** depend on $|\mathcal{H}|$, but rather only on the other natural parameters of the problem, notably the parameter N .
2. The regret it guarantees is with respect to a stronger benchmark than the best hypothesis in the base class. It competes with the best convex combination of hypotheses from the base class, rather than the best one.
3. Finally, the regret guarantee is not multiplicatively approximate anymore; it is directly with respect to the best convex combination without the γ factor.

1.2. Related Work

Boosting and ensemble methods are fundamental in all of machine learning and data science, see e.g. (Olshen, 2001) for a retrospective viewpoint and (Schapire & Freund, 2012a) for a comprehensive text.

Originally boosting was studied for statistical learning and the realizable setting. For a framework that encompasses the different boosting methods see (Bruckhim et al., 2020).

More recently boosting was extended to real-valued prediction via the theory of gradient boosting (Friedman, 2002), and to the online setting (Leistner et al., 2009; Chen et al., 2012; 2014; Beygelzimer et al., 2015b;a; Agarwal et al., 2019; Jung et al., 2017; Jung & Tewari, 2018; Bruckhim & Hazan, 2020). In each of these works, a certain component prevents the result from applying to the full OCO setting. Either the weak learner has a parameter $\gamma = 1$, the convex decision sets are restricted, or the loss functions are restricted.

Our work is the first to generalize online boosting to real-valued prediction in the full setting of online convex optimization, that allows for arbitrary convex decision sets and arbitrary approximation factor γ . This requires the use of a new extension operator, defined henceforth.

For extensive background on online learning and online convex optimization see (Hazan, 2019; Shalev-Shwartz et al., 2012).

The contextual experts and bandits problems have been proposed in (Langford & Zhang, 2008) as a decision making framework with large number of policies. In the online setting, several works study the problem with emphasis on efficient algorithms given access to an ERM oracle (Rakhlin & Sridharan, 2016; Syrgkanis et al., 2016b;a; Rakhlin & Sridharan, 2016).

Another related line of work in online learning studies the possibility of α -regret minimization given access to approximation oracles with an approximate weak learner (Kakade et al., 2009; Hazan et al., 2018). This weaker notion of α -regret only generates approximately optimal decisions. In contrast, our result gives a stronger guarantee of standard regret, and the regret is compared to a stronger benchmark of convex combination of experts.

2. Improper learning and Extension operator

2.1. Preliminaries

For a convex set \mathcal{K} and scalar $c > 0$ we denote by $c\mathcal{K}$ the set of all scaled points, i.e.

$$c\mathcal{K} = \{c\mathbf{x} \text{ s.t. } \mathbf{x} \in \mathcal{K}\}.$$

The algorithmic primitives introduced next use a smoothing operator. Let f be G -Lipschitz continuous. The smoothing operator may be defined via inf-convolution or Moreau-Yoshida regularization:

$$M_\delta[f] = \inf_{y \in \mathbb{R}^d} \left\{ f(y) + \frac{1}{2\delta} \|x - y\|^2 \right\}.$$

The following lemma elucidates well known properties of such an operator (see (Beck, 2017)).

Lemma 3. *The smoothed function $\hat{f}_\delta = M_\delta[f]$ satisfies:*

1. \hat{f}_δ is $\frac{1}{\delta}$ -smooth, and G -Lipschitz.
2. $|\hat{f}_\delta(\mathbf{x}) - f(\mathbf{x})| \leq \frac{\delta G^2}{2}$ for all $\mathbf{x} \in \mathbb{R}^d$.

2.2. The extension operator

The main difficulty we face is that the weak learner only has a γ -approximate regret guarantee. To cope with this, the algorithm we describe henceforth scales the predictions returned by the weak learner by a factor of $\frac{1}{\gamma}$. This algorithm, a rescaled variant of the Frank-Wolfe method, guarantees competitiveness with the convex hull of the base class, if we could somehow play actions in $\frac{1}{\gamma}\mathcal{K}$, instead of \mathcal{K} . Since these actions do not belong to the decision set, they are infeasible.

This presents a significant challenge. Ideally, we would like that for every action in $\gamma^{-1}\mathcal{K}$, one could find an action in \mathcal{K} of comparable (or lesser) loss (for all loss functions). It can be seen that some natural families of functions, i.e. linear functions, do not admit such an operation. For linear loss functions, for example, extrema always occur on the boundary of the set.

To remedy this, we modify the loss function being fed to the Frank-Wolfe algorithm in the first place. We call this

modification an *extension* of the loss function outside \mathcal{K} . Define $\mathbf{Dist}(\mathbf{x}, \mathcal{K})$ the Euclidean distance to \mathcal{K} , and the Euclidean projection $\Pi_{\mathcal{K}}$ as

$$\begin{aligned} \mathbf{Dist}(\mathbf{x}, \mathcal{K}) &= \min_{\mathbf{y} \in \mathcal{K}} \|\mathbf{y} - \mathbf{x}\|, \\ \Pi_{\mathcal{K}}(\mathbf{x}) &= \arg \min_{\mathbf{y} \in \mathcal{K}} \|\mathbf{y} - \mathbf{x}\| \end{aligned}$$

Definition 4 ($(\mathcal{K}, \delta, \kappa)$ -extension). The extension operator over $\mathcal{K} \subseteq \mathbb{R}^d$ is defined as:

$$X_{\mathcal{K}, \delta, \kappa}[f](x) = M_\delta[f(x) + \kappa \cdot \mathbf{Dist}(\mathbf{x}, \mathcal{K})].$$

We henceforth denote by G an upper bound on the norm of the gradients of the loss functions over \mathcal{K} , and set $\kappa = G = \max_t \max_{\mathbf{x} \in \mathbb{R}^d} \|\nabla f_t(\mathbf{x})\|$. The important take-away from these operators is the following lemma, whose importance is crucial in the OCO boosting algorithm 1. The extended loss functions are smooth, agree with the original loss on \mathcal{K} , and possess the property that projection of any action in \mathcal{R}^d onto \mathcal{K} does not increment the extended loss value. In short, it permits projections of infeasible points that are obtained from the weak learners to the feasible domain without an increase in cost.

Lemma 5. *The $(\mathcal{K}, \delta, \kappa)$ -extension of a G -lipschitz function $\hat{f} = X_{\mathcal{K}, \delta, \kappa}[f]$ satisfies the following:*

1. For every point $\mathbf{x} \in \mathcal{K}$, we have $|\hat{f}(\mathbf{x}) - f(\mathbf{x})| \leq \frac{\delta G^2}{2}$.
2. The projection of a point \mathbf{x} onto \mathcal{K} does not increase the $(\mathcal{K}, \kappa, \delta)$ -extended function value by more than $G^2\delta$.

$$\hat{f}(\Pi_{\mathcal{K}}(\mathbf{x})) \leq \hat{f}(\mathbf{x}) + G^2\delta.$$

Proof. Since $\mathbf{Dist}(\mathbf{x}, \mathcal{K}) = 0$ for all $x \in \mathcal{K}$, the first claim follows immediately from the properties of the smoothing operator as stated in Lemma 3. Denote $\mathbf{x}_\pi = \Pi_{\mathcal{K}}(\mathbf{x})$.

$$\begin{aligned} \hat{f}(\mathbf{x}_\pi) - \hat{f}(\mathbf{x}) &\leq f(\mathbf{x}_\pi) - f(\mathbf{x}) - \kappa \cdot \mathbf{Dist}(\mathbf{x}, \mathcal{K}) + G^2\delta \\ &= f(\mathbf{x}_\pi) - f(\mathbf{x}) - \kappa \|\mathbf{x} - \mathbf{x}_\pi\| + G^2\delta \\ &\leq G \|\mathbf{x} - \mathbf{x}_\pi\| - \kappa \|\mathbf{x} - \mathbf{x}_\pi\| + G^2\delta = G^2\delta \end{aligned}$$

Here the first inequality follows from Lemma 3.2, and the second follows from the Lipschitz assumptions on f . The last equality results from the choice $\kappa = G$. \square

3. Algorithm and Main Theorems

Algorithm 1 efficiently converts a weak online learning algorithm into an OCO algorithm with vanishing regret in a black-box manner. The idea is to apply the weak learning algorithm on linear functions that are gradients of the loss.

Algorithm 1 BoOCO = Boosting Online Convex Opt.

- 1: Input: N copies of the γ -WOCL $\mathcal{W}^1, \mathcal{W}^2, \dots, \mathcal{W}^N$, parameters $(\eta_1, \dots, \eta_N), \delta, \kappa = G$.
- 2: **for** $t = 1$ **to** T **do**
- 3: Receive context c_t .
- 4: Choose $\mathbf{x}_t^0 \in \mathcal{K}$ arbitrarily.
- 5: **for** $i = 1$ **to** N **do**
- 6: Define $\mathbf{x}_t^i = (1 - \eta_i)\mathbf{x}_t^{i-1} + \eta_i \frac{1}{\gamma} \mathcal{W}^i(c_t)$.
- 7: **end for**
- 8: Predict $\mathbf{x}_t = \Pi_{\mathcal{K}}(\mathbf{x}_t^N)$ and suffer loss $f_t(\mathbf{x}_t)$.
- 9: Obtain loss function f_t , create $\hat{f}_t = X_{\mathcal{K}, \delta, \kappa}[f_t]$.
- 10: **for** $i = 1$ **to** N **do**
- 11: Pass to \mathcal{W}^i the linear loss function f_t^i .

$$f_t^i(\mathbf{x}) = \nabla \hat{f}_t(\mathbf{x}_t^{i-1}) \cdot \mathbf{x}.$$

- 12: **end for**
- 13: **end for**

The algorithm then recursively applies another weak learner on the gradients of the residual loss, and so forth.

The main performance guarantee we prove is summarized in the following theorem.

Theorem 6 (Main). *The actions \mathbf{x}_t generated by Algorithm 1 with parameter $\delta = \sqrt{\frac{D^2}{\gamma N}}$, $\eta_i = \min\{\frac{2}{i}, 1\}$ satisfy*

$$\begin{aligned} & \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{h \in CH(\mathcal{H})} \sum_{t=1}^T f_t(h(c_t)) \\ & \leq \frac{4GDT}{\gamma\sqrt{N}} + \frac{2GD}{\gamma} \text{Regret}_T(\mathcal{W}). \end{aligned}$$

4. Analysis

Define $\hat{f}_t = X[f_t] = M_\delta[f_t + G \cdot \text{Dist}(\mathbf{x}, \mathcal{K})]$. We apply the setting of $\kappa = G$, as required by Lemma 5, and by Lemma 3, \hat{f}_t is $\frac{1}{3}$ -smooth.

Let $h^* = \arg \min_{h \in CH(\mathcal{H})} \sum_{t=1}^T f_t(h(c_t))$ be the best hypothesis in the convex hull in hindsight. We define $\mathbf{x}_t^* = h^*(c_t)$ as the decisions of this hypothesis.

The main crux of the proof is given by the following lemma.

Lemma 7. *Suppose \hat{f}_t 's are β -smooth, and \hat{G} lipschitz. Then,*

$$\sum_{t=1}^T \hat{f}_t(\mathbf{x}_t^N) - \sum_{t=1}^T \hat{f}_t(\mathbf{x}_t^*) \leq \frac{2\beta D^2 T}{\gamma^2 N} + \frac{\hat{G}D}{\gamma} \text{Regret}_T(\mathcal{W}).$$

Proof. Define for all $i = 0, 1, 2, \dots, N$,

$$\Delta_i = \sum_{t=1}^T \left(\hat{f}_t(\mathbf{x}_t^i) - \hat{f}_t(\mathbf{x}_t^*) \right).$$

Recall that \hat{f}_t is β smooth by our assumption. Therefore:

$$\begin{aligned} \Delta_i &= \sum_{t=1}^T \left[\hat{f}_t(\mathbf{x}_t^{i-1} + \eta_i(\frac{1}{\gamma} \mathcal{W}^i(c_t) - \mathbf{x}_t^{i-1})) - \hat{f}_t(\mathbf{x}_t^*) \right] \\ &\leq \sum_{t=1}^T \left[\hat{f}_t(\mathbf{x}_t^{i-1}) - \hat{f}_t(\mathbf{x}_t^*) + \eta_i \nabla \hat{f}_t(\mathbf{x}_t^{i-1}) \cdot (\frac{1}{\gamma} \mathcal{W}^i(c_t) - \mathbf{x}_t^{i-1}) + \frac{\eta_i^2 \beta}{2} \|\frac{1}{\gamma} \mathcal{W}^i(c_t) - \mathbf{x}_t^{i-1}\|^2 \right] \end{aligned}$$

By using the definition and linearity of f_t^i , we have

$$\begin{aligned} \Delta_i &\leq \sum_{t=1}^T [\hat{f}_t(\mathbf{x}_t^{i-1}) - \hat{f}_t(\mathbf{x}_t^*) \\ &\quad + \eta_i (f_t^i(\frac{1}{\gamma} \mathcal{W}^i(c_t)) - f_t^i(\mathbf{x}_t^{i-1})) + \frac{\eta_i^2 \beta D^2}{2\gamma^2}] \\ &= \Delta_{i-1} + \sum_{t=1}^T \eta_i (\frac{1}{\gamma} f_t^i(\mathcal{W}^i(c_t)) - f_t^i(\mathbf{x}_t^{i-1})) \\ &\quad + \sum_{t=1}^T \frac{\eta_i^2 \beta D^2}{2\gamma^2} \end{aligned}$$

Now, note the following equivalent restatement of the WOCL guarantee, which again utilizes linearity of f_t^i to conclude: linear loss on a convex combination of a set is equal to the same convex combination of the linear loss applied to individual elements.

$$\begin{aligned} & \frac{1}{\gamma} \sum_{t=1}^T f_t^i(\mathcal{W}^i(c_t)) \\ & \leq \min_{h \in \mathcal{H}} \sum_{t=1}^T f_t^i(h(c_t)) + \frac{\hat{G}D \text{Regret}_T(\mathcal{W})}{\gamma} \\ & = \min_{h \in CH(\mathcal{H})} \sum_{t=1}^T f_t^i(h(c_t)) + \frac{\hat{G}D \text{Regret}_T(\mathcal{W})}{\gamma} \end{aligned}$$

Using the above and that $h^* \in CH(\mathcal{H})$, we have

$$\begin{aligned} \Delta_i &\leq \Delta_{i-1} + \sum_{t=1}^T [\eta_i \nabla \hat{f}_t(\mathbf{x}_t^{i-1}) \cdot (\mathbf{x}_t^* - \mathbf{x}_t^{i-1}) \\ &\quad + \frac{\eta_i^2 \beta D^2}{2\gamma^2}] + \eta_i \frac{\hat{G}D}{\gamma} \text{Regret}_T(\mathcal{W}) \\ &\leq \Delta_{i-1} (1 - \eta_i) + \frac{\eta_i^2 \beta D^2 T}{2\gamma^2} + \eta_i R_T \end{aligned}$$

where the last inequality uses the convexity of \hat{f}_t and $R_T = \frac{\hat{G}D}{\gamma} \text{Regret}_T(\mathcal{W})$. We thus have the recurrence

$$\Delta_i \leq \Delta_{i-1} (1 - \eta_i) + \eta_i^2 \frac{\beta D^2 T}{2\gamma^2} + \eta_i R_T.$$

Denoting $\hat{\Delta}_i = \Delta_i - R_T$,

$$\hat{\Delta}_i \leq \hat{\Delta}_{i-1}(1 - \eta_i) + \eta_i^2 \frac{\beta D^2 T}{2\gamma^2}.$$

Lemma 8. Let $\{h_t\}$ be a sequence that satisfies the recurrence

$$h_{t+1} \leq h_t(1 - \eta_t) + \eta_t^2 c.$$

Then taking $\eta_t = \min\{1, \frac{2}{t}\}$ implies

$$h_t \leq \frac{4c}{t}.$$

This is a recursive formula that can be solved by applying Lemma 8; we obtain that $\hat{\Delta}_N \leq \frac{2\beta D^2 T}{\gamma^2 N}$. \square

We are ready to prove the main guarantee of Algorithm 1.

Proof of Theorem 6. Using both parts of Lemma 5 in succession, we have

$$\begin{aligned} & \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_t^*) \\ & \leq \sum_{t=1}^T \hat{f}_t(\mathbf{x}_t) - \sum_{t=1}^T \hat{f}_t(\mathbf{x}_t^*) + \delta G^2 T \\ & \leq \sum_{t=1}^T \hat{f}_t(\mathbf{x}_t^N) - \sum_{t=1}^T \hat{f}_t(\mathbf{x}_t^*) + 2\delta G^2 T \end{aligned}$$

Next, recall by Lemma 3, that \hat{f}_t is $\frac{1}{\delta}$ -smooth. By applying Lemma 7, and optimizing δ , we have

$$\begin{aligned} & \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_t^*) \\ & \leq 2\delta G^2 T + \frac{2D^2 T}{\delta\gamma^2 N} + \frac{\hat{G}D}{\gamma} \text{Regret}_T(\mathcal{W}) \\ & = \frac{4GDT}{\gamma\sqrt{N}} + \frac{\hat{G}D}{\gamma} \text{Regret}_T(\mathcal{W}) \end{aligned}$$

Lemma 9. $\text{Dist}(\mathbf{x}, \mathcal{K})$ is 1-Lipschitz,

Using the above lemma, we claim that $\hat{G} \leq 2G$. Therefore, using Lemma 3, $\|\nabla f_t^i(x_t^i)\| = \|\nabla \hat{f}_t(x_t^i)\| \leq 2G$. \square

Proof of Lemma 9. Observe the following.

$$\begin{aligned} & \text{Dist}(\mathbf{x}, \mathcal{K}) - \text{Dist}(\mathbf{y}, \mathcal{K}) \\ & = \|\mathbf{x} - \Pi_{\mathcal{K}}(\mathbf{x})\| - \|\mathbf{y} - \Pi_{\mathcal{K}}(\mathbf{y})\| \\ & \leq \|\mathbf{x} - \Pi_{\mathcal{K}}(\mathbf{y})\| - \|\mathbf{y} - \Pi_{\mathcal{K}}(\mathbf{y})\| \quad \Pi_{\mathcal{K}}(\mathbf{y}) \in \mathcal{K} \\ & \leq \|\mathbf{x} - \mathbf{y}\| \quad \text{triangle inequality} \end{aligned}$$

\square

Proof of Lemma 8. This is proved by induction on t .

Induction base. For $t = 1$, we have

$$h_2 \leq h_1(1 - \eta_1) + \eta_1^2 c = c \leq 4c$$

Induction step.

$$\begin{aligned} h_{t+1} & \leq (1 - \eta_t)h_t + \eta_t^2 c \\ & \leq \left(1 - \frac{2}{t}\right) \frac{4c}{t} + \frac{4c}{t^2} \quad \text{induction hypothesis} \\ & = \frac{4c}{t} \left(1 - \frac{1}{t}\right) \\ & \leq \frac{4c}{t} \cdot \frac{t}{t+1} \quad \frac{t-1}{t} \leq \frac{t}{t+1} \\ & = \frac{4c}{t+1} \end{aligned}$$

\square

5. Boosting for Bandit Linear Optimization

Recall that the setting of Bandit Linear Optimization (BLO) is exactly the same as OCO, but (1) the cost functions are linear, and (2) the feedback for the decision maker is the cost it incurs (no gradient information). We use the notation that our linear cost function at time t is:

$$f_t(\mathbf{x}) = f_t^\top \mathbf{x}, \quad f_t \in \mathbb{R}^d.$$

To boost in the linear bandit setting, we apply techniques for unbiased gradient estimation. Namely, we use randomness to create a random linear function whose expectation is the actual linear function. We can then hope to use the previous algorithm on the random linear functions and obtain a probabilistic regret bound.

For the rest of this section, we assume the following,

Assumption 1. The convex decision set \mathcal{K} contains the unit simplex $K \supseteq \Delta_d$.

This assumption is very unrestrictive: by scaling and rotation, any set \mathcal{K} which is non-degenerate contains the simplex. Scaling changes regret by a constant factor, and thus we scale so as to contain the unit simplex for notational convenience. An axis-aligning rotation does not affect the regret bound. We use standard basis aligned simplex for notational simplicity.

We can now use standard randomized exploration to create an unbiased estimator of the loss function, as per the following scheme. Let b_t be a Bernoulli random variable with parameter η . Let i_t be chosen uniformly at random from $i_t \in \{1, 2, \dots, d\}$.

$$\tilde{f}_t(i) = \begin{cases} 0 & i \neq i_t \\ 0 & i = i_t, b_t = 0 \\ \frac{d}{\eta} \times f_t(i_t) & i = i_t, b_t = 1 \end{cases} \quad (2)$$

Clearly the expectation of the random vector \tilde{f}_t equals $\mathbb{E}[\tilde{f}_t] = f_t$. We can now use this random variable as feedback for Algorithm 1, as given in Algorithm 2.

Algorithm 2 BoBLO = Boosting Bandit Linear Opt.

- 1: Input: parameters η , Algorithm 1 instance \mathcal{A} .
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Observe context c_t .
 - 4: Draw random Bernoulli b_t w. parameter η .
 - 5: **if** $b_t = 1$ **then**
 - 6: Pick a coordinate basis $i_t \in \mathcal{R}^d$ vector randomly.
 - 7: Play $\mathbf{x}_t = i_t$.
 - 8: Pass \tilde{f}_t as per Equation (2) to \mathcal{A} .
 - 9: **else**
 - 10: Play $\mathbf{x}_t = \mathcal{A}(c_t)$, and pass the $\mathbf{0}$ loss vector to \mathcal{A} .
 - 11: **end if**
 - 12: **end for**
-

Notice that Algorithm 2 calls upon weak learners for OCO with full information. For this algorithm, our main performance guarantee is as follows. The resulting bound is sub-linear whenever the full-information regret bound is sublinear. Therefore, boosting in bandit linear settings is feasible whenever the full-information version is feasible, albeit at a slower rate.

Theorem 10 (Boosting for Bandit Linear Optimization). *The predictions \mathbf{x}_t generated by Algorithm 2 satisfy*

$$\begin{aligned} & \mathbb{E} \left[\sum_{t=1}^T f_t(\mathbf{x}_t) \right] - \inf_{h \in CH(\mathcal{H})} \sum_{t=1}^T f_t(h(c_t)) \\ & \leq GD \sqrt{T \left(\frac{4dT}{\gamma\sqrt{N}} + \frac{2d}{\gamma} \text{Regret}_T(\mathcal{W}) \right)} \end{aligned}$$

Proof. Let $\mathbf{y}_t = \mathcal{A}(c_t)$. Observe that

$$\begin{aligned} & \mathbb{E} \left[\sum_{t=1}^T f_t(\mathbf{x}_t) \right] - \inf_{h \in CH(\mathcal{H})} \sum_{t=1}^T f_t(h(c_t)) \\ & \leq \mathbb{E} \left[\sum_{t=0}^T (\tilde{f}_t(\mathbf{y}_t) - \tilde{f}_t(\mathbf{x}_t^*)) \right] + \eta GDT \\ & \leq \frac{4D\tilde{G}T}{\gamma\sqrt{N}} + \frac{2D\tilde{G}}{\gamma^2} \text{Regret}_T(\mathcal{W}) + \eta T \quad \text{Theorem 6} \\ & \leq \frac{16DdGT}{\eta\gamma\sqrt{N}} + \frac{8dDG}{\eta\gamma} \text{Regret}_T(\mathcal{W}) + \eta T \end{aligned}$$

Above, we use \tilde{G} for an upper bound on the gradients of \tilde{f}_t . Lastly, by construction the gradients of the loss function \tilde{f}_t are bounded by $\frac{dG}{\eta}$. Balancing η concludes the claim \square

5.1. Application to contextual bandits

The contextual bandits problem is one of the most well studied special cases of contextual BLO. In multi-armed

contextual bandits, the decision set is a set of K discrete arms, the losses are $[0, 1]$ -bounded, and the feedback available to learner is solely the loss of the arm it picks. This is exactly BLO for the decision set $\mathcal{K} = \Delta_K$. We state a corollary of our result for this setting.

Corollary 11 (Multi-armed Contextual Bandits). *The predictions \mathbf{x}_t generated by Algorithm 2 with the decision set $\mathcal{K} = \Delta_K$ and the loss set $[0, 1]^K$ satisfy*

$$\begin{aligned} & \mathbb{E} \left[\sum_{t=1}^T f_t(\mathbf{x}_t) \right] - \inf_{h \in \mathcal{H}} \sum_{t=1}^T f_t(h(c_t)) \\ & \leq K \sqrt{T \left(\frac{4T}{\gamma\sqrt{N}} + \frac{2}{\gamma} \text{Regret}_T(\mathcal{W}) \right)} \end{aligned}$$

6. Boosting for Stochastic Contextual Optimization

In this section we give an alternative viewpoint of our boosting results from the lens of stochastic contextual optimization. The mathematical development is analogous to the previous results, but the statement of results may be of independent interest in statistical learning theory and contextual learning.

Let $\mathcal{K} \subseteq \mathbb{R}^d$ be the (convex) decision set, and \mathcal{C} be the set of possible contexts. In the Stochastic Contextual Optimization problem the aggregate loss attributable to a hypothesis is the expected cost of its actions with respect to a joint distribution \mathcal{D} over the context set \mathcal{C} and the set of convex cost functions over the decision set \mathcal{K} .

$$F_{\mathcal{D}}(h) = \mathbb{E}_{(f,c) \sim \mathcal{D}} [f(h(c))].$$

Instead of explicit optimization over the hypothesis class, we assume that we have access to experts that are approximately competitive with the base hypothesis class. More formally, we define a weak optimizer as follows.

Definition 12. A γ -weak contextual optimizer is a learning algorithm that, when given samples from a distribution \mathcal{D} over unit linear loss functions and contexts, outputs a mapping $\mathcal{W} : \mathcal{C} \rightarrow \mathcal{K}$ such that

$$F_{\mathcal{D}}(\mathcal{W}) \leq \gamma \cdot \min_{h \in \mathcal{H}} F_{\mathcal{D}}(h) + \varepsilon.$$

Typically ε scales as $\sim \frac{1}{\sqrt{m}}$ when the learner is given m samples from the distribution \mathcal{D} .

The results in this section do **not** follow blackbox from the previous ones. While online boosting algorithms operate in more permissive settings than statistical ones, as they do not require the context and loss vectors to be sampled i.i.d., the corresponding assumption on the weak learners for the online setting are stronger too.

6.1. Algorithm and main theorem

Algorithm 3 below makes use of N weak optimizers, and generates a hypothesis $h : \mathcal{C} \rightarrow \mathcal{K}$ such that the following theorem holds. Notice that the resulting guarantee delivers a solution which is stronger in two aspects than the weak optimization oracle:

1. The resulting solution competes with the convex hull of \mathcal{H} , rather than the best fixed $h \in \mathcal{H}$.
2. The resulting solution removes the γ approximation factor of the weak optimizers.

Theorem 13 (Main-SCO). *The hypothesis h generated by Algorithm 3 with parameter $\delta = \sqrt{\frac{D^2}{\gamma N}}$ satisfy*

$$F_{\mathcal{D}}(h) - \inf_{h^* \in CH(\mathcal{H})} F_{\mathcal{D}}(h^*) \leq \frac{4GD}{\gamma\sqrt{N}} + \frac{2GD}{\gamma}\varepsilon.$$

Algorithm 3 B4CO = Boosting for Contextual Opt.

- 1: Input: γ -weak contextual optimizer O , parameters $\{\eta_1, \dots, \eta_N\}$, $\delta, \kappa = G$, distribution \mathcal{D} .
 - 2: Choose h^0 arbitrarily from \mathcal{H} .
 - 3: **for** $i = 1$ **to** N **do**
 - 4: Define a distribution \mathcal{D}^i specified via the following sampling oracle:
 - Sample $(f, c) \sim \mathcal{D}$, and define

$$\hat{f} = X_{\mathcal{K}, \delta, \kappa}[f]$$
 - Define a linear loss function

$$f^i(\mathbf{x}) = \nabla \hat{f}(h^{i-1}(c)) \cdot \mathbf{x}$$
 - Output (f^i, c) .
 - 5: Call a weak optimizer on the distribution \mathcal{D}^i to get a hypothesis \mathcal{W}^i .
 - 6: Define $h^i = (1 - \eta_i)h^{i-1} + \frac{\eta_i}{\gamma}\mathcal{W}^i$
 - 7: **end for**
 - 8: Return predictor h , defined as $h(c) = \Pi_{\mathcal{K}}(h^N(c))$.
-

As before, for a convex loss function f , define $\hat{f} = X_{\mathcal{K}, \delta, G}[f]$, and note that \hat{f}_t is $\frac{1}{\delta}$ -smooth. Notations we use below:

1. $F(h) = \mathbb{E}_{(f,c) \sim \mathcal{D}}[f(h(c))]$
2. $\hat{F}(h) = \mathbb{E}_{(f,c) \sim \mathcal{D}}[\hat{f}(h(c))]$

Lemma 14. *Suppose, for any $f \in \mathcal{F}$, \hat{f} is β -smooth, and $\max_{f \in \mathcal{F}, x \in \mathcal{K}} \|\nabla \hat{f}(x)\| \leq \hat{G}$. Then,*

$$\hat{F}(h) - \hat{F}(h^*) \leq \frac{2\beta D^2}{\gamma^2 N} + \frac{\hat{G}D}{\gamma}\varepsilon.$$

Proof. Let $h^* = \arg \min_{h \in CH(\mathcal{H})} F(h)$. Denote for all $i = 0, 1, 2, \dots, N$, $\Delta_i = \hat{F}(h^i) - \hat{F}(h^*)$.

Recall that \hat{f}_t is β smooth by our assumption. Thus,

$$\begin{aligned} \Delta_i &= \hat{F}(h^i) - \hat{F}(h^*) \\ &= \mathbb{E}_{f,c \sim \mathcal{D}}[\hat{f}(h^i(c))] - \hat{F}(h^*) \\ &= \mathbb{E}_{f,c \sim \mathcal{D}} \left[\hat{f}(h^{i-1}(c)) + \eta_i \left(\frac{1}{\gamma} \mathcal{W}^i(c) - h^{i-1}(c) \right) \right] \\ &\quad - F(h^*) \\ &\leq \mathbb{E}_{f,c \sim \mathcal{D}}[\hat{f}_t(h^{i-1}(c))] \\ &\quad + \eta_i \nabla \hat{f}(h^{i-1}(c)) \cdot \left(\frac{1}{\gamma} \mathcal{W}^i(c) - h^{i-1}(c) \right) \\ &\quad + \frac{\eta_i^2 \beta}{2} \left\| \frac{1}{\gamma} h^{i-1}(c) - \mathcal{W}^i(c) \right\|^2 - \hat{F}(h^*) \\ &\leq \mathbb{E}_{f,c \sim \mathcal{D}}[\hat{f}(h^{i-1}(c))] \\ &\quad + \eta_i \left(f^i \left(\frac{1}{\gamma} \mathcal{W}^i(c) \right) - f_t^i(h^{i-1}(c)) \right) \\ &\quad + \frac{\eta_i^2 \beta D^2}{2\gamma^2} - \hat{F}(h^*) \end{aligned}$$

Using linearity of f^i , which implies that optimal aggregate loss values are equal over \mathcal{H} and $CH(\mathcal{H})$ hypotheses classes, and the definition of weak learning, we have

$$\begin{aligned} \Delta_i &\leq \Delta_{i-1} \\ &\quad + \eta_i \mathbb{E}_{f,c \sim \mathcal{D}} \left[\nabla \hat{f}_t(h^{i-1}(c)) \cdot (h^*(c) - h^{i-1}(c)) \right] \\ &\quad + \frac{\eta_i^2 \beta D^2}{2\gamma^2} + \eta_i \frac{\hat{G}D\varepsilon}{\gamma} \\ &\leq \Delta_{i-1} (1 - \eta_i) + \frac{\eta_i^2 \beta D^2}{2\gamma^2} + \eta_i \frac{\hat{G}D\varepsilon}{\gamma} \end{aligned}$$

where the last step follows from the convexity of f . We thus have the recurrence

$$\Delta_i \leq \Delta_{i-1} (1 - \eta_i) + \frac{\eta_i^2 \beta D^2}{2\gamma^2} + \eta_i \frac{\hat{G}D\varepsilon}{\gamma}.$$

As before, substituting $\hat{\Delta}_i = \Delta_i - \frac{\hat{G}D\varepsilon}{\gamma}$, and applying Lemma 8, gives us $\hat{\Delta}_N \leq \frac{2\beta D^2}{\gamma^2 N}$. \square

Proof of Theorem 13. Using Lemma 5, we have,

$$\begin{aligned} F(h) - F(h^*) &= \mathbb{E}_{f,c}[f(h(c))] - \mathbb{E}_{f,c}[f(h^*(c))] \\ &\leq \mathbb{E}_{f,c}[\hat{f}(h(c))] - \mathbb{E}_{f,c}[\hat{f}(h^*(c))] + \delta G^2 \\ &\leq \mathbb{E}_{f,c}[\hat{f}(h^N(c))] - \mathbb{E}_{f,c}[\hat{f}(h^*(c))] + 2\delta G^2 \end{aligned}$$

Boston dataset						
	WL	N=2	N=3	N=4	N=5	Improvement
Decision Stumps	1.000	0.941	0.891	0.853	0.821	17.9%
Ridge Regression	1.000	0.980	0.985	1.009	1.060	2.0%
Tiny MLP	1.000	0.987	0.966	0.934	0.920	8.0%
Diabetes dataset						
	WL	N=2	N=3	N=4	N=5	Improvement
Decision Stumps	1.000	0.975	0.960	0.952	0.941	15.9%
Ridge Regression	1.000	0.986	0.974	0.965	0.956	4.4%
Tiny MLP	1.000	0.982	0.981	0.987	0.965	3.5%
California Housing dataset						
	WL	N=2	N=3	N=4	N=5	Improvement
Decision Stumps	1.000	0.969	0.946	0.932	0.922	7.8%
Ridge Regression	1.000	1.019	1.041	1.066	1.094	-1.9%
Tiny MLP	1.000	0.860	0.861	0.888	0.835	16.5%

Figure 1. Performance of the boosting algorithm on 3 different datasets, averaged across 20 runs, as a function of the weak learning class and the number of weak learners. The entries indicate normalized loss value with the base/weak learner loss set to one.

By Lemma 3, that \hat{f} is $\frac{1}{\delta}$ -smooth. By applying Lemma 14:

$$F(\mathcal{A}) - F(h^*) \leq 2\delta G^2 + \frac{2D^2}{\delta\gamma^2 N} + \frac{\hat{G}D}{\gamma}\varepsilon$$

Because $\text{Dist}(x, K)$ is 1-Lipschitz, using Lemma 3, $\|\nabla f_t^i(x_t^i)\| = \|\nabla \hat{f}_t(x_t^i)\| \leq 2G$. Choosing the suggested value of δ yields the proof. \square

7. Experimental Results

While the primary contribution of this work is theoretical, empirically testing our proposal serves as a sanity check for the theoretical results. Also, as stated in the related work, the proposed algorithm is the first of its kind in terms of being able to simultaneously operate with general convex losses and sets, and with multiplicatively approximate (inexact) weak learners. Consequently, the role of experiments here is to corroborate the theory, and is not aimed at achieving state-of-the-art on the considered datasets.

To validate our results, we check if the proposed algorithm is indeed capable of boosting the accuracy of concrete instantiations of weak learners. Three online weak learners were considered²: decision stumps, ridge regression, and a tiny multi-layer perceptron with one hidden unit trained via online gradient descent.

For each choice of weak learner class, we considered the performance of the boosting algorithm (Algorithm 1) across multiple rounds of boosting or number of weak learners; the computational burden of the algorithm scales linearly with the latter. We evaluated the weak learners and the boosting

²We use machine learning models from Scikit-Learn (Pedregosa et al., 2011) to implement base learners.

algorithm on 3 publicly available datasets³ for regression with square loss, averaging each across 20 runs. The step size η was set to 0.01, and γ was chosen to be 0.1 – these values were not tuned. We present the results in Table 1. Further experimental details and code are detailed in the supplement.

The results demonstrate the boosting for online convex optimization does indeed succeed in boosting the accuracy while using only few weak learners (equivalently, within a few rounds of boosting), and therefore, with a reasonably small increase in computation. The improvement is most pronounced for decision stumps, since this is an especially impoverished base class. Also, note that a convex combination of linear models is linear. Therefore, boosting a linear model (like ridge regression) has no expressivity benefit. Mirroring this, the improvements in the ridge regression setting are least prominent, and in some cases negative.

8. Conclusions

We study the problem of online convex optimization in the contextual setting. To cope with the computational challenges, we take a boosting approach, and generalize the methodology of gradient boosting and online boosting to contextual online convex optimization.

Our derivation introduces a new extension operator for convex functions over convex domains which may be of independent interest.

³The California Housing dataset (Pace & Barry, 1997) and the Boston dataset (Harrison Jr & Rubinfeld, 1978) can be found at <http://lib.stat.cmu.edu/datasets/>. The Diabetes dataset used here is present in UCI ML Repository (<https://archive.ics.uci.edu/ml/datasets/Diabetes>).

Acknowledgements

Elad Hazan was supported in part by National Science Foundation Award 1704860.

References

- Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646, 2014.
- Agarwal, N., Brukhim, N., Hazan, E., and Lu, Z. Boosting for dynamical systems. *arXiv preprint arXiv:1906.08720*, 2019.
- Arora, S., Hazan, E., and Kale, S. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.
- Beck, A. *First-order methods in optimization*. SIAM, 2017.
- Beygelzimer, A., Hazan, E., Kale, S., and Luo, H. Online gradient boosting. In *Advances in neural information processing systems*, pp. 2458–2466, 2015a.
- Beygelzimer, A., Kale, S., and Luo, H. Optimal and adaptive algorithms for online boosting. In *International Conference on Machine Learning*, pp. 2323–2331, 2015b.
- Brukhim, N. and Hazan, E. Online boosting with bandit feedback. *arXiv preprint arXiv:2007.11975*, 2020.
- Brukhim, N., Chen, X., Hazan, E., and Moran, S. Online agnostic boosting via regret minimization, 2020.
- 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 (JACM)*, 44(3):427–485, 1997.
- Chen, S.-T., Lin, H.-T., and Lu, C.-J. An online boosting algorithm with theoretical justifications, 2012.
- Chen, S.-T., Lin, H.-T., and Lu, C.-J. Boosting with online binary learners for the multiclass bandit problem. In *International Conference on Machine Learning*, pp. 342–350, 2014.
- Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.
- Friedman, J. H. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.
- Harrison Jr, D. and Rubinfeld, D. L. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.
- Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.
- Hazan, E. and Koren, T. The computational power of optimization in online learning. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 128–141, 2016.
- Hazan, E., Hu, W., Li, Y., and Li, Z. Online improper learning with an approximation oracle. In *Advances in Neural Information Processing Systems*, pp. 5652–5660, 2018.
- Jung, Y. H. and Tewari, A. Online boosting algorithms for multi-label ranking. In *International Conference on Artificial Intelligence and Statistics*, pp. 279–287, 2018.
- Jung, Y. H., Goetz, J., and Tewari, A. Online multiclass boosting. In *Advances in neural information processing systems*, pp. 919–928, 2017.
- Kakade, S. M., Kalai, A. T., and Ligett, K. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.
- Langford, J. and Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pp. 817–824, 2008.
- Leistner, C., Saffari, A., Roth, P. M., and Bischof, H. On robustness of on-line boosting—a competitive study. In *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1362–1369. IEEE, 2009.
- Littlestone, N. and Warmuth, M. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- Olshen, R. A conversation with Leo Breiman. *Statistical Science*, 16(2):184–198, 2001.
- Pace, R. K. and Barry, R. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Rakhlin, A. and Sridharan, K. Bistro: An efficient relaxation-based method for contextual bandits. In *ICML*, pp. 1977–1985, 2016.
- Schapire, R. E. and Freund, Y. *Boosting: Foundations and Algorithms*. Cambridge university press, 2012a. ISBN 9780262017183. doi:

10.1017/CBO9781107415324.004. URL https://www.cambridge.org/core/product/identifier/CBO9781107415324A009/type/book_part.

Schapire, R. E. and Freund, Y. *Boosting: Foundations and Algorithms*. MIT Press, 2012b.

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

Syrgkanis, V., Krishnamurthy, A., and Schapire, R. Efficient algorithms for adversarial contextual learning. In *International Conference on Machine Learning*, pp. 2159–2168, 2016a.

Syrgkanis, V., Luo, H., Krishnamurthy, A., and Schapire, R. E. Improved regret bounds for oracle-based adversarial contextual bandits. In *Advances in Neural Information Processing Systems*, pp. 3135–3143, 2016b.

Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pp. I–I. IEEE, 2001.