
Online Learning with Imperfect Hints

Aditya Bhaskara¹ Ashok Cutkosky^{2,3} Ravi Kumar² Manish Purohit²

Abstract

We consider a variant of the classical online linear optimization problem in which at every step, the online player receives a “hint” vector before choosing the action for that round. Rather surprisingly, it was shown that if the hint vector is guaranteed to have a positive correlation with the cost vector, then the online player can achieve a regret of $O(\log T)$, thus significantly improving over the $O(\sqrt{T})$ regret in the general setting. However, the result and analysis require the correlation property at *all* time steps, thus raising the natural question: can we design online learning algorithms that are resilient to bad hints?

In this paper we develop algorithms and nearly matching lower bounds for online learning with imperfect directional hints. Our algorithms are oblivious to the quality of the hints, and the regret bounds interpolate between the always-correlated hints case and the no-hints case. Our results also generalize, simplify, and improve upon previous results on optimistic regret bounds, which can be viewed as an additive version of hints.

1. Introduction

In the standard online convex optimization model (Zinkevich, 2003), at each time step t , an algorithm first plays a point x_t in a convex set, and then the system responds with a convex loss function. The loss incurred by the algorithm is the function evaluated at the point x_t . The performance of an algorithm is measured using the concept of regret. The *regret* of an algorithm is the difference between the total loss it incurs and the loss of the best fixed point it could have played (in hindsight); algorithms with sub-linear regret are

hence desirable. The framework of online convex optimization is quite powerful, general, and has been extensively studied. Many important problems such as portfolio selection, learning from mixture of experts, matrix completion, recommendation systems, and certain online combinatorial optimization problems can be cast in this framework. For a detailed exposition, see the books by Hazan (2016) and Shalev-Shwartz (2011).

An important special case of online convex optimization is when the loss function is actually linear, i.e., the loss function is given by a *cost vector*. In this case, algorithms with regret $O(\sqrt{T})$, where T is the number of steps, are known (Zinkevich, 2003; Kalai & Vempala, 2005); furthermore, this bound is also optimal (Cesa-Bianchi & Lugosi, 2006). In fact, from a regret point of view, the linear case is the hardest since if the loss function is strongly convex, then there are algorithms achieving only $O(\log T)$ regret (Hazan et al., 2007). There has been some effort to better understand the regret landscape of linear loss functions, especially on how to circumvent the pessimistic $\Omega(\sqrt{T})$ barrier.

A particularly intriguing line of work was initiated by Hazan & Megiddo (2007), who modeled a notion of predictability in online learning settings. In their model, the algorithm knows the first coordinate of the cost vector at all time steps. Under this assumption, they showed a regret bound of $O(d^2/\alpha \cdot \log T)$ when the convex set is the Euclidean ball, where α is the magnitude of the first coordinate that is known to the algorithm and d is the dimension of the space. Their work was subsequently generalized and extended by Dekel et al. (2017), who considered a scenario when the online algorithm is provided with a *directional hint* at each step; this hint is assumed to be always weakly but positively correlated with the cost vector. They showed a regret bound of $O(d/\alpha \cdot \log T)$, where α is the amount of correlation present in the hint.

The biggest drawback in these previous works is that they require the hints to be helpful at *every* time step. Clearly, this is a stringent requirement that may easily fail to hold. This is especially so if the hints are provided by, say, a learning algorithm! In such a scenario, one can only expect the hints to be good on average or have other probabilistic guarantees of goodness. This means in particular that some of the hints could potentially be very misleading. Since the

¹University of Utah, Salt Lake City Utah, USA ²Google Research, Mountain View California, USA ³Boston University, Boston Massachusetts, USA. Correspondence to: Aditya Bhaskara <bhaskaraaditya@gmail.com>, Ashok Cutkosky <ashok@cutkosky.com>, Ravi Kumar <ravi.k53@gmail.com>, Manish Purohit <mpurohit@google.com>.

algorithm is oblivious to the quality of each individual hint, it is desirable to have an algorithm that is both consistent and robust: utilize the good hints as well as possible to minimize regret, while at the same time not be damaged too much by bad hints. Specifically, the algorithm should never incur worse than $O(\sqrt{T})$ regret, as otherwise the algorithm was better off not using any hints at all! This type of ML-provided hints and their role in improving combinatorial online algorithms have generated a lot of recent interest for problems such as caching (Lykouris & Vassilvitskii, 2018; Rohatgi, 2020; Kumar et al., 2020), ski-rental (Kumar et al., 2018), bipartite matching (Kumar et al., 2019), and scheduling (Kumar et al., 2018; Lattanzi et al., 2020). This serves as another motivation for our work.

Formulation. We consider the online convex optimization problem with a linear loss function in the presence of hints that can be imperfect. At each time step t , the algorithm is provided with a hint vector h_t . After the algorithm plays a point x_t , a cost vector c_t is revealed and the algorithm incurs a loss of $\langle c_t, x_t \rangle$. The hint vector h_t “typically” gives non-trivial information about c_t . Formally, given a parameter α , a hint h_t is said to be good if it satisfies $\langle c_t, h_t \rangle \geq \alpha \|c_t\|^2$ and bad otherwise.

Our results. We design an algorithm that achieves a regret bound that smoothly interpolates between the two extreme cases when the hints h_t are good at all time steps and when hints are arbitrarily wrong. In particular, for any $\alpha > 0$, we obtain a regret of

$$O\left(\frac{(1 + \sqrt{B})}{\alpha} \log(1 + T - B)\right),$$

where B is the number of times steps when the hints are bad, i.e., $\langle c_t, h_t \rangle < \alpha \|c_t\|^2$. The dependence on B turns out to be nearly optimal as we will show in Section 4. We also generalize these results when the underlying feasible space is (q, μ) -uniformly convex and show matching lower bounds. For the formal statements, see Theorems 3.1 and A.3.

Surprisingly, our algorithm simultaneously also yields improved regret guarantees when the hint h_t is viewed as an additive estimate of the cost vector: a hint is good if $\|c_t - h_t\|$ is small. This notion of hint was considered in Rakhlin & Sridharan (2013); Hazan & Kale (2010); Mohri & Yang (2016); Steinhardt & Liang (2014), who gave regret bounds of the form $O\left(\sqrt{\sum_{t=1}^T \|c_t - h_t\|^2}\right)$. We achieve a regret $\tilde{O}\left(\sqrt{\sum_{t=1}^T (\|c_t - h_t\|^2 - \|h_t\|^2)}\right)$ (see Corollary 3.7).

Even when restricted to the special case where the hints are all good, our result improves upon the regret bound of Dekel et al. (2017) in multiple ways. First, our regret bound is *dimension-free*, i.e., better by a factor of the dimension of

the space. Second, our algorithm is significantly faster: their work relied on expensive matrix calculations yielding $O(d^2)$ computation per round, while our algorithm runs in $O(d)$ time, matching simple gradient descent. Third, our proofs are simpler as we rely on loss functions that are easily seen to be strongly convex (as opposed to proving exp-concavity). Furthermore, for the case of $q > 2$, Dekel et al. (2017) only obtained comparable regret bounds when all the hints are in the same direction. We generalize this in two ways, allowing different hints at each step and a small number of bad hints.

Finally, we consider the *unconstrained* variant of online optimization, where the algorithm allowed to play any point $x_t \in \mathbb{B}$, while achieving a regret that depends on $\|u\|$ for *all* $u \in \mathbb{B}$. This setting is discussed in Section 5.

2. Preliminaries

Let \mathbb{B} be a real Banach space with norm $\|\cdot\|$ and let \mathbb{B}^* be its dual space with norm $\|\cdot\|_*$. Let $\vec{c} = c_1, c_2, \dots$ be cost vectors in \mathbb{B}^* such that $\|c_t\|_* \leq 1$. In the classical online learning setting, c_1, c_2, \dots arrive one by one and at time t , an algorithm \mathcal{A} responds with a vector $x_t \in \mathbb{B}$, *before* c_t arrives. The *regret* of the algorithm \mathcal{A} for a vector $u \in \mathbb{B}$ is

$$\mathcal{R}_{\mathcal{A}}(u, \vec{c}, T) = \sum_{t=1}^T \langle c_t, x_t - u \rangle,$$

where we use the $\langle \cdot, \cdot \rangle$ notation to denote the application of a dual vector in \mathbb{B}^* to a vector in \mathbb{B} . (For instance if \mathbb{B} is the space \mathbb{R}^d with $\|\cdot\|$ being the ℓ_2 -norm, we have $\mathbb{B} = \mathbb{B}^*$ and $\langle \cdot, \cdot \rangle$ will correspond to the standard inner product.)

We consider the case when there are *hints* available to an algorithm. Let $\vec{h} = h_1, h_2, \dots$ be the hints, where each hint $h_t \in \mathbb{B}$, $\|h_t\| \leq 1$, is available to the algorithm \mathcal{A} at time t ; this hint is available *before* \mathcal{A} responds with x_t . The regret definition is the same and is denoted $\mathcal{R}_{\mathcal{A}}(u, \vec{c}, T \mid \vec{h})$.

The hints need not be perfect. To capture this, let $\alpha > 0$ be a fixed *threshold*. We define $G_{T,\alpha}$ to be the set of indices t where the hint h_t is good, i.e., has a large correlation with c_t . Similarly, we define $B_{T,\alpha}$ to be the set of indices where the hint is bad. Formally, we define:

$$G_{T,\alpha} = \{t \leq T : \langle c_t, h_t \rangle \geq \alpha \cdot \|c_t\|_*^2\}, \quad \text{and} \\ B_{T,\alpha} = \{t \leq T : \langle c_t, h_t \rangle < \alpha \cdot \|c_t\|_*^2\}.$$

Let $B_T = B_{T,0}$, i.e., the time steps when h_t is negatively correlated with c_t . We will also use a compressed-sum notation for indexed variables: $a_{1:t} = \sum_{i=1}^t a_i$.

Let $\mathbb{K} = \{x \in \mathbb{B} : \|x\| \leq 1\}$. We consider two settings, a *constrained* setting where we must choose $x_t \in \mathbb{K}$ and an *unconstrained* setting sans this restriction. In the former case, we will be concerned only with bounding $\mathcal{R}_{\mathcal{A}}(u, \vec{c}, T)$ for $u \in \mathbb{K}$, while in the latter we will consider any $u \in \mathbb{B}$.

Finally, we establish some notation about convex functions

and spaces. For a convex function f , we use $\partial f(x) \subset \mathbb{B}^*$ to denote the set of *subgradients* of f at x . We say that f is μ -strongly convex with respect to the norm $\|\cdot\|$ if for all x, y and $g \in \partial f(x)$, we have $f(y) \geq f(x) + \langle g, y - x \rangle + \frac{\mu}{2}\|x - y\|^2$. We say that the Banach space \mathbb{B} is μ -strongly convex if the function $\frac{1}{2}\|x\|^2$ is μ -strongly convex with respect to $\|\cdot\|$ for some $\mu > 0$. We note this notion is equivalent to the definition of strong convexity of a space used in Dekel et al. (2017); e.g., see the discussion after Definition 4.16 in Pisier (2016). Further, a Banach space is *reflexive* if the natural injection $i : \mathbb{B} \rightarrow \mathbb{B}^{**}$ given by $\langle i(x), c \rangle = \langle c, x \rangle$ is an isomorphism of Banach spaces. Note that all finite-dimensional Banach spaces are reflexive. Throughout this paper, we assume that \mathbb{B} is reflexive and μ -strongly convex.

A typical example is $\mathbb{B} = \mathbb{R}^d$ with $\|\cdot\|$ equal to the standard ℓ_2 norm. In this case \mathbb{B} is reflexive and 1-strongly convex.

3. Constrained Learning with Imperfect Hints

We first consider the *constrained* setting of the problem in which the online algorithm must choose a point $x_t \in \mathbb{K}$ at all time steps $t \leq T$. To illustrate our main ideas, we first focus on the case when the Banach space \mathbb{B} is μ -strongly convex. Our techniques also extend to general (q, μ) -uniformly convex spaces and we present this extension in Appendix A.

Theorem 3.1. *Consider the online linear optimization problem over a Banach space with a μ -strongly convex norm, where at every step we receive a hint vector h_t and need to output a point $x_t \in \mathbb{K}$. Then there is an efficient algorithm that for any $\alpha > 0$, achieves regret*

$$O\left(\sqrt{\sum_{t \in B_{T,\alpha}} \|c_t\|_*^2} + \frac{r_T}{\mu\alpha} \log\left(1 + \sum_{t \in G_{T,\alpha}} \|c_t\|_*^2\right)\right),$$

where $r_T = \sqrt{1 + \sum_{t \in B_T} |\langle c_t, h_t \rangle|}$.

We remark about the order of quantifiers in the theorem. The bound holds for any $\alpha > 0$ and the algorithm itself is oblivious to α . Thus, if we have B bad hints (i.e., $|B_{T,\alpha}| = B$), then $r_T \leq \sqrt{1 + B}$ and the number of good steps is $T - B$, so we obtain the upper bound of $O(\frac{\sqrt{1+B}}{\alpha} \log(1 + T - B))$. Also, the bound is never larger than \sqrt{T} , because if α is large, $G_{T,\alpha} = \emptyset$, and thus the first term is the only one that remains, and it is $\leq \sqrt{T}$.

Outline of the algorithm. Our algorithm (denoted ALG) can be best viewed as a procedure that interacts with an “inner” online convex optimization subroutine, which we denote by \mathcal{A} . At every step, ALG receives a prediction \bar{x}_t from \mathcal{A} , which it modifies using the hint h_t , and produces x_t . Then the algorithm receives c_t , using which it produces a function ℓ_t (which depends on h_t, c_t , and an additional parameter r_t that ALG maintains). This function, along with relevant parameters, are passed to \mathcal{A} . The key properties that

Algorithm 1 OLO with imperfect hints (Procedure ALG)

input Hints h_t followed by cost vectors c_t

Define $\lambda_0 = 1/\mu$ and $r_0 = 1$.

for $t = 1 \dots T$ **do**

 Get hint h_t

 Get \bar{x}_t from procedure \mathcal{A} , and set

$$x_t = \bar{x}_t + \frac{(\|\bar{x}_t\|^2 - 1)}{2r_t} h_t$$

 Play x_t and receive cost $c_t \in \mathbb{B}^*$

if $\langle c_t, h_t \rangle < 0$ **then**

 Set $r_{t+1} = \sqrt{r_t^2 + |\langle c_t, h_t \rangle|}$

else

 Set $r_{t+1} = r_t$

end if

 Define $\sigma_t = \frac{|\langle c_t, h_t \rangle| \mu}{r_t}$

 Define λ_t as the solution to:

$$\lambda_t = \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu\lambda_{1:t}}$$

 Send the loss function $\ell_{h_t, r_t, c_t}(x)$, λ_t to procedure \mathcal{A}

 // (loss function defined in (1))

end for

Algorithm 2 FTRL with adaptive rate (Procedure \mathcal{A})

input Convex functions ℓ_t , parameters λ_t

At $t = 1$ return $\bar{x}_1 = 0$

for $t = 2 \dots T$ **do**

 Output $\bar{x}_t := \operatorname{argmin}_{\|x\| \leq 1} \ell_{1:t-1}(x) + \frac{\lambda_{0:t-1}}{2} \|x\|^2$

 Receive loss ℓ_t and parameter λ_t

end for

we show are: (a) the regret of ALG can be related to the regret of the procedure \mathcal{A} , and (b) the functions ℓ_t are strongly convex, and thus the regret of \mathcal{A} can be bounded efficiently using known techniques. The parameter r_t encapsulates the “confidence in hints” seen so far.

Algorithms 1 and 2 describe the procedures ALG and \mathcal{A} . Intuitively, given a prediction \bar{x}_t , we should be able to improve the loss $\langle c_t, x_t \rangle$ by playing instead $x_t = \bar{x}_t - h_t$; assuming the hint h_t is positively correlated with c_t . However, there are two immediate problems with this approach. First, if h_t is negatively correlated with c_t then we have actually increased the loss. Second, this addition operation may cause x_t to leave the set \mathbb{K} , which is not allowed. We address both concerns by setting $x_t = \bar{x}_t - \delta_{r_t}(x_t) h_t$, where $\delta_{r_t}(x_t) = \frac{1 - \|x\|^2}{2r_t}$ is a carefully chosen scale factor.

The *surrogate loss function* used in the algorithm is:

$$\ell_{h_t, r_t, c_t}(x) = \langle c_t, x \rangle + \frac{|\langle c_t, h_t \rangle|}{2r_t} (\|x\|^2 - 1). \quad (1)$$

It is clear from the description that as the algorithm proceeds,

r_t is monotone increasing and hence $r_t \geq 1$ for all t . We first demonstrate that Algorithm 1 always plays a feasible point, i.e., $x_t \in \mathbb{K}$ for all t .

Lemma 3.2. *For any t , $\|x_t\| \leq 1$. In other words, the point x_t played by Algorithm 1 is always feasible.*

Proof. From the description of \mathcal{A} , $\|\bar{x}_t\| \leq 1$. Thus since $r_t \geq 1$ and by the triangle inequality, we have

$$\begin{aligned} \|x_t\| &\leq \|\bar{x}_t\| + \frac{(1 - \|\bar{x}_t\|^2)}{2} \|h_t\| \\ &\leq \|\bar{x}_t\| + \frac{(1 - \|\bar{x}_t\|^2)}{2} \\ &= \|\bar{x}_t\| + \frac{(1 - \|\bar{x}_t\|)(1 + \|\bar{x}_t\|)}{2}. \end{aligned}$$

This is clearly ≤ 1 , as $\|\bar{x}_t\| \leq 1$. \square

We next establish some basic properties of the surrogate loss function.

Lemma 3.3. *Let ℓ_t denote ℓ_{h_t, r_t, c_t} defined in (1). This function satisfies:*

1. If $\langle c_t, h_t \rangle \geq 0$, then $\ell_t(\bar{x}_t) = \langle c_t, x_t \rangle$.
2. If $\langle c_t, h_t \rangle < 0$, then

$$\langle c_t, x_t \rangle \leq \ell_t(\bar{x}_t) + \frac{|\langle c_t, h_t \rangle|}{r_t}.$$

3. For all $u \in \mathbb{B}$ with $\|u\| \leq 1$, $\ell_t(u) \leq \langle c_t, u \rangle$.
4. $\ell_t(x)$ is $\frac{|\langle c_t, h_t \rangle| \mu}{r_t}$ -strongly convex.
5. $\ell_t(x)$ is $2 \|c_t\|_*$ -Lipschitz.

Proof. The first three properties are immediate from the definitions of ℓ_t , x_t and the fact that $\|\bar{x}_t\| \leq 1$ and $r_t \geq 1$. The fourth one follows from the fact that $\frac{1}{2} \|x\|^2$ is μ -strongly convex, and that adding a convex function to a strongly convex function preserves strong convexity. The last property is also a consequence of the fact that $\|x\|^2$ is 2-Lipschitz inside the unit ball (which follows from $\|x\|^2 - \|y\|^2 = (\|x\| + \|y\|)(\|x\| - \|y\|)$) and since $r_t \geq 1$. \square

This implies the following lemma, which is crucial for our argument. It relates the regret of ALG with the regret of FTRL (procedure \mathcal{A}). Recall the definition of B_T from before (the time steps when the hints are negatively correlated with the cost vector).

Lemma 3.4. *Let $u \in \mathbb{B}$ satisfy $\|u\| \leq 1$, and let ℓ_t be shorthand for ℓ_{h_t, r_t, c_t} as before. Then*

$$\mathcal{R}_{\text{ALG}}(u, \vec{c}, T) \leq \mathcal{R}_{\mathcal{A}}(u, \vec{\ell}, T) + \sum_{t \in B_T} \frac{|\langle c_t, h_t \rangle|}{r_t}. \quad (2)$$

Proof. By definition, $\mathcal{R}_{\text{ALG}}(u, \vec{c}, T) = \sum_t \langle c_t, x_t \rangle - \langle c_t, u \rangle \leq \sum_t \langle c_t, x_t \rangle - \ell_t(u)$, by Property 3 in Lemma 3.3.

Now using the first two properties, we have that when the hints are positively correlated, i.e., $\langle c_t, h_t \rangle \geq 0$, we have $\langle c_t, x_t \rangle = \ell_t(\bar{x}_t)$, and otherwise (i.e., $t \in B_T$) we have $\langle c_t, x_t \rangle \leq \ell_t(\bar{x}_t) + \frac{|\langle c_t, h_t \rangle|}{r_t}$. This completes the proof of the lemma. \square

We bound the first term in (2) using known results for FTRL, and the second term by the following simple lemma.

Lemma 3.5. *From our definition of r_t , we have*

$$\sum_{t \in B_T} \frac{|\langle c_t, h_t \rangle|}{r_t} \leq 2 \sqrt{\sum_{t \in B_T} |\langle c_t, h_t \rangle|}.$$

Proof. From our algorithm, note that r_t is precisely $\sqrt{1 + \sum_{\tau < t, \tau \in B_T} |\langle c_\tau, h_\tau \rangle|}$. Thus, since all the terms $|\langle c_t, h_t \rangle|$ are ≤ 1 , we can use the fact that for all non-negative real numbers $\{z_i\}_{i=1}^m$,

$$\sum_{t=1}^m \frac{z_t}{\sqrt{z_{1:t}}} \leq 2\sqrt{z_{1:m}},$$

to the numbers $|\langle c_t, h_t \rangle|$ for $t \in B_T$. This implies the lemma. (The fact above is standard in the analysis of FTRL; for instance, see Lemma 4 of McMahan (2017).) \square

It remains to bound the regret of the FTRL procedure \mathcal{A} . We now use the general techniques presented in McMahan (2017); Hazan et al. (2008) to do this.

Lemma 3.6. *Suppose we run procedure \mathcal{A} using our choice of $\ell_t, \lambda_t, \sigma_t$. Then for any $\alpha > 0$ and $\|u\| \leq 1$, the regret $\mathcal{R}_{\mathcal{A}}(u, \vec{\ell}, T)$ is at most*

$$\frac{1}{2\mu} + 4 \left(\frac{\sqrt{\sum_{t \in B_T, \alpha} \|c_t\|_*^2}}{\mu} + \frac{r_T \log(1 + \mu \sum_{t \in G_T, \alpha} \|c_t\|_*^2)}{\alpha \mu} \right),$$

where $r_T = \sqrt{1 + \sum_{t \in B_T} |\langle c_t, h_t \rangle|}$.

Proof. Note that $\ell_t = \ell_{h_t, r_t, c_t}$ is σ_t -strongly convex as we observed earlier, so that the function $\ell_{1:t}(x) + \frac{\lambda_{0:t-1}}{2} \|x\|^2$ is $\sigma_{1:t} + \mu \lambda_{0:t-1}$ -strongly convex. Then, using the analysis of the FTRL procedure (Theorem 1 of McMahan (2017)), we set g_t to be an arbitrary subgradient of ℓ_t at \bar{x}_t and obtain:

$$\mathcal{R}_{\mathcal{A}}(u, \vec{\ell}, T) \leq \frac{\lambda_{0:T}}{2} \|u\|^2 + \frac{1}{2} \sum_t \frac{\|g_t\|_*^2}{\sigma_{1:t} + \mu \lambda_{0:t-1}}$$

Since ℓ_t is $2 \|c_t\|_*$ -Lipschitz (Lemma 3.3), we have that $\|g_t\|_*^2 \leq 4 \|c_t\|_*^2$, so the regret is:

$$\leq \frac{\lambda_0}{2} + 2 \left(\lambda_{1:T} + \sum_t \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu \lambda_{0:t-1}} \right)$$

Next, observe that since $\|c_t\|_* \leq 1$, we must have $\lambda_t \leq \frac{1}{\mu} = \lambda_0$ for all t . Therefore the regret is

$$\leq \frac{1}{2\mu} + 2 \left(\sum_{t=1}^T \lambda_t + \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu\lambda_{1:t}} \right). \quad (3)$$

Now, we can use our choice of λ_t to appeal to the result of Hazan et al. (2008); see Lemma 3.1 of their paper. We also reproduce a slightly more general version of this result in Lemma B.10 for completeness. This lets us replace our choice of λ_t with *any other choice* up to constants, yielding:

$$\mathcal{R}_{\mathcal{A}}(u, \vec{\ell}, T) \leq \frac{1}{2\mu} + 4 \cdot \min_{\lambda_t^*} \left\{ \sum_{t=1}^T \lambda_t^* + \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu\lambda_{0:t}^*} \right\}.$$

Let us now show how to pick λ_t^* that depend on the parameter $\alpha > 0$, thus giving the bound in the lemma. Define $Q_\alpha = \sum_{t \in B_{T,\alpha}} \|c_t\|_*^2$, i.e., the total squared norm at time steps where the desired correlation condition between the hint and the cost vector is not met. Now set $\lambda_1^* = \sqrt{1 + Q_\alpha}$ and $\lambda_t^* = 0$ for $t > 1$. Then $\mathcal{R}_{\mathcal{A}}(u, \vec{\ell}, T)$ is at most:

$$\frac{1}{2\mu} + 4 \left(\sqrt{1 + Q_\alpha} + \sum_{t=1}^T \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu\sqrt{1 + Q_\alpha}} \right).$$

We can separate the sum into $t \in B_{T,\alpha}$ and indices outside (i.e., in $G_{T,\alpha}$). This gives:

$$\sum_{t=1}^T \frac{\|c_t\|_*^2}{\sigma_{1:t} + \mu\sqrt{1 + Q_\alpha}} \leq \frac{Q_\alpha}{\mu\sqrt{1 + Q_\alpha}} + \sum_{t \in G_{T,\alpha}} \frac{\|c_t\|_*^2}{1 + \sigma_{1:t}}.$$

The first term is clearly $\leq \sqrt{Q_\alpha}/\mu$. To analyze the second term, we use the fact that for any $t \in G_{T,\alpha}$, we have

$$\sigma_t \geq \frac{\alpha\mu \|c_t\|_*^2}{r_t} \geq \frac{\alpha\mu \|c_t\|_*^2}{r_T},$$

where in the last step we used the monotonicity of r_t . Thus by denoting the numbers $\{\|c_t\|_*^2\}_{t \in G_{T,\alpha}}$ by w_1, w_2, \dots, w_m (in order), we have

$$\begin{aligned} \sum_{t \in G_{T,\alpha}} \frac{\|c_t\|_*^2}{1 + \sigma_{1:t}} &\leq \frac{r_T}{\alpha\mu} \sum_{i \in [m]} \frac{w_i}{\frac{r_T}{\alpha\mu} + w_{1:i}} \\ &\leq \frac{r_T}{\alpha\mu} \int_{r_T/\alpha\mu}^{w_{1:m} + (r_T/\alpha\mu)} \frac{dz}{z}. \end{aligned}$$

Since $\frac{r_T}{\alpha\mu} \geq \frac{1}{\mu}$, we can bound this by $\frac{r_T}{\alpha\mu} \log(1 + \mu w_{1:m})$. Recalling the definition of r_T , the proof follows. \square

Theorem 3.1 now follows immediately from Lemmas 3.4, 3.5, and 3.6.

Remark. The regret bound in Theorem 3.1 has two important terms. The first term depends on the sum of the squared norm of the cost vectors over all the time indices $t \in B_{T,\alpha}$ when the hint vector was not strongly correlated with the cost. As we show in Section 4, such a dependence is unavoidable. The second term is

$$\begin{aligned} &\frac{1}{\alpha} \cdot \sqrt{1 + \sum_{t \in B_T} |\langle c_t, h_t \rangle|} \log \left(1 + \mu \sqrt{\sum_{t \in G_{T,\alpha}} \|c_t\|_*^2} \right) \\ &\leq \frac{\sqrt{1 + |B_T|}}{\alpha} \log(1 + \mu |G_{T,\alpha}|). \end{aligned}$$

In the special case when all hints are α -correlated, we have $|B_T| = |B_{T,\alpha}| = 0$ and $|G_{T,\alpha}| = T$, which improves upon regret bounds of Dekel et al. (2017) since we drop the dependence on the dimension.

In Appendix A, we show that our algorithm directly extends to the case when the underlying Banach space \mathbb{B} is (q, μ) -uniformly convex for $q > 2$ to yield a regret bound of $O\left(T^{\frac{q-2}{q-1}}\right)$.

3.1. Recovering and improving optimistic bounds

In this section we relate our notion of hints in the constrained setting to the idea of *optimistic* regret. For simplicity, we focus on the case that \mathbb{B} is a Hilbert space and $\|\cdot\|$ is the Hilbert space norm (or, for concreteness, that $\mathbb{B} = \mathbb{R}^d$ and $\|\cdot\|$ is the ℓ_2 norm). In this setting we can write $\mathbb{B} = \mathbb{B}^*$ and $\|\cdot\| = \|\cdot\|_*$. Recall that prior optimistic algorithms (e.g., (Rakhlin & Sridharan, 2013)) achieve regret bounds of the form:

$$\mathcal{R}(u, \vec{c}, T) = O \left(\sqrt{\sum_{t=1}^T \|c_t - h_t\|^2} \right).$$

Interestingly, in the *unconstrained* case, Cutkosky (2019) achieves regret

$$\tilde{O} \left(\sqrt{\max \left(1, \sum_{t=1}^T \|c_t - h_t\|^2 - \|h_t\|^2 \right)} \right),$$

which sacrifices a logarithmic factor to improve $\|c_t - h_t\|^2$ to $\|c_t - h_t\|^2 - \|h_t\|^2$. However, their construction failed to achieve such a result when there are constraints. Here, we show that in fact our same algorithm *with no modifications* obtains this refined optimistic bound when constrained to the unit ball. Specifically, we have the following result:

Corollary 3.7. *Let \mathbb{B} be a Hilbert space. Then Algorithm 1 guarantees regret on the unit ball \mathbb{K} :*

$$\frac{1}{2} + \left(8 + 16 \log(1 + T) \right) \sqrt{Z}.$$

where $Z = 1 + \sum_{t=1}^T \max(\|c_t - h_t\|^2 - \|h_t\|^2, 0)$.

Proof. Recall that in a Hilbert space, $\mu = 1$ and $q = p = 2$.

Then, looking at the regret bound of Theorem 3.1, we have

$$\begin{aligned} \mathcal{R}_{\text{ALG}}(u, \vec{c}, T) &\leq \frac{1}{2} + 4 \sqrt{\sum_{t \in B_{T,\alpha}} \|c_t\|^2} \\ &\quad + 2r_T + \frac{4r_T}{\alpha} \log \left(1 + \sum_{t \in G_{T,\alpha}} \|c_t\|^2 \right), \end{aligned}$$

where $r_T = \sqrt{1 + \sum_{t \in B_T} |\langle c_t, h_t \rangle|}$.

Next, notice that for any $t \in B_T$, we have

$$\begin{aligned} |\langle c_t, h_t \rangle| &= -\langle c_t, h_t \rangle \\ &\leq \frac{1}{2} \|c_t\|^2 - \langle c_t, h_t \rangle \leq \frac{1}{2} (\|c_t - h_t\|^2 - \|h_t\|^2). \end{aligned}$$

Therefore,

$$r_T \leq \sqrt{1 + \frac{1}{2} \sum_{t=1}^T \max(\|c_t - h_t\|^2 - \|h_t\|^2, 0)}.$$

Further, if we set $\alpha = \frac{1}{4}$, then for any $t \in B_{T,\alpha}$, we have

$$\begin{aligned} \|c_t\|^2 &\leq \|c_t\|^2 + \|c_t\|^2 - 4\langle c_t, h_t \rangle \\ &= 2(\|c_t - h_t\|^2 - \|h_t\|^2). \end{aligned}$$

Therefore,

$$\sum_{t \in B_{T,\alpha}} \|c_t\|^2 \leq \sqrt{2 \sum_{t=1}^T \max(\|c_t - h_t\|^2 - \|h_t\|^2, 0)}.$$

Putting all this together and over-approximating constants, we can conclude the proof. \square

4. Lower Bounds

We now show that the regret bounds achieved by our algorithms are near-optimal. Recall that the regret bound had two terms: one corresponding to hints that are negatively correlated with c_t , and one corresponding to hints that are positively correlated, but not ‘‘correlated enough’’. Our first lower bound shows that even the second term is necessary.

4.1. Bad hints are uncorrelated

Assume that we are in Euclidean space with the standard ℓ_2 norm, where the algorithm needs to play a point in the unit ball. We show the following:

Theorem 4.1. *There exists a sequence of hint vectors h_1, h_2, \dots and cost vectors c_1, c_2, \dots with the following properties: (a) $\langle c_t, h_t \rangle \geq 0$ for all t , (b) for all but B time steps, we have $\langle c_t, h_t \rangle = \|c_t\|$ (i.e., hints are perfect), and (c) any online learning algorithm that plays given the hints incurs an expected regret of $\Omega(\sqrt{B})$.*

Proof. We consider the following example in two dimensions, with orthogonal unit vectors e_1 and e_2 . For the first

B time steps, suppose that $h_t = e_2$, and $c_t = \pm e_1$, where the sign is chosen uniformly at random at each step. Now, let $z = c_1 + \dots + c_B$. For the rest of the time steps, suppose that $h_t = c_t = z / \|z\|$. In other words, we have the standard one-dimensional ‘‘hard instance’’ in the first B steps (which incurs an expected regret of \sqrt{B}), appended with time steps where the hints are perfect.

Any online algorithm incurs an expected loss 0 on the first B steps (and loss $-(t - B)$ on the rest of the steps), while we have the expected $\|z\| = \sqrt{B}$, and so playing the vector $-z / \|z\|$ at all the time steps incurs a total loss of $-(t - B) - \sqrt{B}$. Thus the expected regret is \sqrt{B} . \square

The proof above (as well as the ones that follow) exhibit a distribution over instances for which any deterministic algorithm incurs an expected regret of \sqrt{B} . Applying Yao’s lemma (e.g., (Motwani & Raghavan, 1995)), the regret lower bound therefore applies to randomized algorithms as well.

4.2. Bad hints are spread out over time

Theorem 4.1 is taking advantage of an adversarial distribution of bad hints. By placing all the useless hints at the beginning of the game, we force the algorithm to experience high regret that it cannot recover from. It turns out such overtly adversarial distributions are not necessary: even if the bad hints are randomly distributed, the algorithm must still suffer high regret. (We note that in this case, we no longer have $\langle c_t, h_t \rangle \geq 0$ for all rounds.)

Theorem 4.2. *Consider the one-dimensional problem with domain being the unit interval $[-1, 1]$. Suppose $h_t = 1$ for all t and that each c_t takes value $p - 1$ with probability p and value p with probability $1 - p$, for $p = B/T$ and $B \leq T/4$. Then the expected number of bad hints is B and the expected regret of any algorithm is at least $\sqrt{B}/2$.*

Proof. Note that a hint is negatively correlated with the cost if the cost is negative, which happens with probability p . Thus the expected number of bad hints is $pT = B$. Now at each step, we have $\mathbb{E}[c_t] = 0$. Thus, whatever x_t the algorithm plays, we have that $\mathbb{E}[\sum_{t=1}^T c_t x_t] = 0$; thus, the expected loss of the algorithm is 0. Finally, we have that the vector $z = c_{1:t}$ has norm $\mathbb{E}[\|z\|] = \sqrt{p(1-p)T} \geq \sqrt{B}/2$. Therefore compared to the best vector in hindsight, namely $-\frac{z}{\|z\|}$, the expected regret is at least $\sqrt{B}/2$. \square

4.3. A lower bound for the ℓ_q norm

Next, we show that even when the hint is always $\Omega(1)$ correlated with the cost, our upper bound for general q (which is $T^{(q-2)/(q-1)}$) is optimal in the class of dimension-free bounds.

Theorem 4.3. *There exists a sequence of hints h_1, h_2, \dots*

and costs c_1, c_2, \dots in \mathbb{R}^{T+1} such that (a) $\langle c_t, h_t \rangle \geq \Omega(1)$ for all t , and (b) any online learning algorithm that plays given the hints incurs an expected regret of $\Omega\left(T^{\frac{(q-2)}{(q-1)}}\right)$.

Proof. Let $e_0, e_1, e_2, \dots, e_T$ be orthogonal, unit length vectors in our space, and suppose that at time $t = 1, 2, 3, \dots$, we have $c_t = e_0 \pm e_t$, where the sign is chosen u.a.r. (to keep all the vectors of $\|\cdot\| \leq 1$, we can normalize c_t ; this does not change the analysis, so we skip this step).

Now, suppose the algorithm plays vectors x_1, x_2, \dots . We have the total expected loss to be exactly $\sum_t \langle e_0, x_t \rangle$, which has magnitude at most T . Let us construct a vector u with $\|u\|_q \leq 1$ that has a higher magnitude for the inner product. Let us denote $z = c_{1:T} = T e_0 + \sum_{t=1}^T \sigma_t e_t$, for some signs σ_t . Define $u = \sum_{t=0}^T \beta_t u_t$, where:

$$\beta_0 = 1 - \frac{3}{2q} T^{-\frac{1}{q-1}}; \quad \beta_t = \sigma_t T^{-\frac{1}{q-1}}.$$

We have $\sum_{t=1}^T \beta_t^q = T \cdot T^{-\frac{q}{q-1}} = T^{-\frac{1}{q-1}}$. Next, we make the simple observation that for any $\gamma < 1/2$, $(1 - \frac{3\gamma}{2q})^q \leq e^{-3\gamma/2} \leq 1 - \gamma$. Using this, we have $\beta_0^q \leq 1 - T^{-\frac{1}{q-1}}$, and thus $\|u\|_q \leq 1$. Next, we have

$$\begin{aligned} \langle z, u \rangle &= \left(1 - \frac{3}{2q} T^{-\frac{1}{q-1}}\right) T + T \cdot T^{-\frac{1}{q-1}} \\ &= T + \left(1 - \frac{3}{2q}\right) T^{1-\frac{1}{q-1}}. \end{aligned}$$

Thus, compared to the point $-u$, any algorithm has an expected regret $T^{(q-2)/(q-1)}$. This completes the proof. \square

Indeed, even if we allow a dependence on dimension, obtaining a $\log T$ regret is impossible for $q > 2$. We refer to Appendix C for a regret lower bound (quite similar to the above) even in two dimensions. In this case, the lower bound interpolates between $\log T$ (which we achieve for $q = 2$) and \sqrt{T} (which is achievable if we lose a factor linear in the dimension).

5. Unconstrained Learning with Hints

We now consider the unconstrained setting where the online algorithm is allowed to output any $x \in \mathbb{B}$. In this section, we show that the unconstrained setting is much simpler than the constrained version of the problem.

Recall the definition of $B_{T,\alpha}$. For the unconstrained setting, we work with a more relaxed notion of bad hints. Let $B_{T,\alpha}^*$ be the smallest set of indices such that $\sum_{t \in [T] \setminus B_{T,\alpha}^*} \langle c_t, h_t \rangle \geq \alpha \cdot \sum_{t \in [T] \setminus B_{T,\alpha}^*} \|c_t\|^2$. We observe that, by definition, for any $\alpha > 0$, we have $|B_{T,\alpha}^*| \leq |B_{T,\alpha}|$.

Our algorithm is essentially a black-box reduction to a standard parameter-free online linear optimization algorithm

without any hints and follows the framework of adding independent online learning algorithms (Cutkosky, 2019). In fact, our algorithm is identical to the optimistic online learning algorithm of Cutkosky (2019). However, we are able to obtain better regret guarantees by a tighter analysis.

Denote $C_T = \sum_{t=1}^T \|c_t\|^2$.

Lemma 5.1. *Let \mathcal{A} be a parameter-free online linear optimization algorithm that guarantees a regret bound of:*

$$\mathcal{R}_{\mathcal{A}}(u, \vec{c}, T) \leq f(\|u\|, C_T, \epsilon), \quad \forall \epsilon > 0,$$

for some function $f(\cdot, \cdot, \cdot)$ where $f(0, \cdot, \epsilon) \leq \epsilon$ and f is monotone in all the three parameters. Then, there exists an algorithm \mathcal{B} for online learning with hints that guarantees the regret bound:

$$\begin{aligned} \mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) &\leq \min \left\{ f(\|u\|, C_T, \epsilon) + \epsilon, \right. \\ &\quad \left. \inf_{0 \leq y \leq \|u\|} \left\{ 2f(\|u\|, C_T, \epsilon) - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\} \right\}. \end{aligned}$$

Proof. We design an algorithm \mathcal{B} that utilizes the provided online learning algorithm \mathcal{A} in two distinct settings. First, let $x_t \in \mathbb{B}$ be the output of algorithm \mathcal{A} in response to loss vectors $c_1, \dots, c_{t-1} \in \mathbb{B}^*$. We also use algorithm \mathcal{A} in the scalar (i.e., \mathbb{R}) setting by providing $-\langle c_t, h_t \rangle$ as the losses. Let y_t be the output of algorithm \mathcal{A} in response to loss vectors $-\langle c_1, h_1 \rangle, \dots, -\langle c_t, h_t \rangle$.

On receiving hints h_1, \dots, h_t , and losses for the previous time steps c_1, \dots, c_{t-1} , our algorithm \mathcal{B} outputs

$$z_t = x_t - y_t h_t.$$

Then for all $u \in \mathbb{B}$, we have

$$\begin{aligned} \mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) &= \sum_{t=1}^T \langle c_t, z_t - u \rangle \\ &= \sum_{t=1}^T \langle c_t, x_t - u \rangle - \sum_{t=1}^T y_t \langle c_t, h_t \rangle \\ &= \inf_{y \in \mathbb{R}} \left\{ \sum_{t=1}^T \langle c_t, x_t - u \rangle + \sum_{t=1}^T \langle c_t, h_t \rangle (y - y_t) \right. \\ &\quad \left. - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\} \\ &\leq \inf_{y \in \mathbb{R}} \left\{ f(\|u\|, C_T, \epsilon) + f(|y|, \sum_{t=1}^T \langle c_t, h_t \rangle^2, \epsilon) \right. \\ &\quad \left. - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\}, \end{aligned}$$

using the regret bounds guaranteed by algorithm \mathcal{A} . Setting $y = 0$ is sufficient to obtain the first part of the regret bound. To obtain the second part of the bound, we use

$\langle c_t, h_t \rangle^2 \leq \|c_t\|_*^2$ and the monotonicity of f to obtain

$$\mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) \leq \inf_{0 \leq y \leq \|u\|} \left\{ 2f(\|u\|, C_T, \epsilon) - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\}.$$

□

We are now ready to present our main result for unconstrained online learning with hints.

Theorem 5.2. *For the unconstrained online linear optimization problem with hints, for any $\alpha > 0$, there exists an algorithm \mathcal{B} that guarantees for any $u \in \mathbb{B}$ and $\epsilon > 0$, we have $\mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) \leq$*

$$\tilde{O} \left(\epsilon + \frac{\|u\| \log \left(1 + \frac{T\|u\|}{\epsilon^2} \right) \left(1 + \sqrt{|B_{T,\alpha}^*|} \right)}{\alpha \mu} \right).$$

Proof. An algorithm \mathcal{A} that satisfies the properties of Lemma 5.1 is provided by Cutkosky & Orabona (2018). Their algorithm guarantees

$$f(\|u\|, C_T, \epsilon) = \epsilon + 8\|u\| \log \left(\frac{8\|u\|^2(1+4C_T)^{4.5}}{\epsilon^2} + 1 \right) + \frac{4\|u\|}{\sqrt{\mu}} \sqrt{C_T \left(2 + \log \left(\frac{5\|u\|^2(2+8C_T)^9}{\epsilon^2} + 1 \right) \right)}.$$

Similar algorithms with differing constants and dependencies on the c_t are described in Jun & Orabona (2019); Orabona & Pál (2016); Cutkosky & Sarlos (2019); McMahan & Orabona (2014); Foster et al. (2018; 2017); Kempka et al. (2019); van der Hoeven (2019).

Applying Lemma 5.1 with this algorithm \mathcal{A} , we get

$$\begin{aligned} \mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) &\leq \inf_{0 \leq y \leq \|u\|} \left\{ 2f(\|u\|, C_T, \epsilon) - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\} \\ &= \inf_{\substack{y \leq \|u\| \\ y \geq 0}} \left\{ 2\epsilon + Q_1 + \|u\| Q_2 \sqrt{C_T} - y \sum_{t=1}^T \langle c_t, h_t \rangle \right\}, \end{aligned} \quad (4)$$

where we let $Q_1 = 16\|u\| \log \left(\frac{8\|u\|^2(1+4C_T)^{4.5}}{\epsilon^2} + 1 \right)$ and $Q_2 = \frac{8}{\sqrt{\mu}} \sqrt{\left(2 + \log \left(\frac{5\|u\|^2(2+8C_T)^9}{\epsilon^2} + 1 \right) \right)}$ for brevity.

However, by definition of $B_{T,\alpha}^*$, we have

$$\begin{aligned} \sum_{t=1}^T \langle c_t, h_t \rangle &= \sum_{t \in [T] \setminus B_{T,\alpha}^*} \langle c_t, h_t \rangle + \sum_{t \in B_{T,\alpha}^*} \langle c_t, h_t \rangle \\ &\geq \alpha \sum_{t=1}^T \|c_t\|^2 + \sum_{t \in B_{T,\alpha}^*} (\langle c_t, h_t \rangle - \alpha \|c_t\|^2) \\ &\geq \alpha \sum_{t=1}^T \|c_t\|^2 - 2|B_{T,\alpha}^*|. \end{aligned}$$

Substituting back into (4) and using $y = \|u\|/\sqrt{|B_{T,\alpha}^*|}$,

$$\mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h}) \leq 2\epsilon + Q_1 + \|u\| Q_2 \sqrt{C_T} - \frac{\alpha \|u\|}{\sqrt{|B_{T,\alpha}^*|}} C_T + 2\|u\| \sqrt{|B_{T,\alpha}^*|}. \quad (5)$$

However for any C_T , we have

$$Q_2 \sqrt{C_T} - \frac{\alpha}{\sqrt{|B_{T,\alpha}^*|}} C_T \leq \frac{Q_2^2 \sqrt{|B_{T,\alpha}^*|}}{4\alpha};$$

indeed, this follows since $\left(Q_2 - 2 \frac{\alpha}{\sqrt{|B_{T,\alpha}^*|}} \sqrt{C_T} \right)^2 \geq 0$.

And thus (5) yields that $\mathcal{R}_{\mathcal{B}}(u, \vec{c}, T \mid \vec{h})$ is at most:

$$\begin{aligned} &\leq 2\epsilon + Q_1 + \|u\| \frac{Q_2^2 \sqrt{|B_{T,\alpha}^*|}}{4\alpha} + 2\|u\| \sqrt{|B_{T,\alpha}^*|} \\ &= 2\epsilon + 16\|u\| \log \left(\frac{8\|u\|^2(1+4C_T)^{4.5}}{\epsilon^2} + 1 \right) \\ &\quad + \frac{64\|u\| \left(2 + \log \left(\frac{5\|u\|^2(2+8C_T)^9}{\epsilon^2} + 1 \right) \right) \sqrt{|B_{T,\alpha}^*|}}{4\alpha \mu} \\ &\quad + 2\|u\| \sqrt{|B_{T,\alpha}^*|} \\ &= \tilde{O} \left(\epsilon + \frac{\|u\| \log \left(1 + \frac{T\|u\|}{\epsilon^2} \right) \left(1 + \sqrt{|B_{T,\alpha}^*|} \right)}{\alpha \mu} \right). \quad \square \end{aligned}$$

The bound of Theorem 5.2 is similar to our results in the constrained setting, but now we have replaced $B_{T,\alpha}$ with the relaxed quantity $B_{T,\alpha}^*$. The unconstrained algorithms requires the good hints to be good only *on average*, while the constrained algorithm required each individual good hint to be good. This is a significant relaxation: consider our lower bound argument of Theorem 4.1, in which $\langle c_t, h_t \rangle$ is 0 for the first $\frac{T}{2}$ rounds and 1 afterwards. A constrained algorithm must suffer $O(\sqrt{T})$ regret in this setting, but in the unconstrained case the hints are $\frac{1}{2}$ -correlated on average, and so the algorithm will suffer only $O(\log T)$ regret. It is *strictly easier* to take advantage of hints in the unconstrained setting than in the constrained setting.

6. Conclusions

In this work we obtained an algorithm for online linear optimization in the presence of imperfect hints. Our algorithm generalizes previous results that used hints in online optimization to get improved regret bounds, but were not robust against hints that were not guaranteed to be good. By tolerating bad hints while getting optimal regret bounds, our work thus makes it possible for the hints to be derived from a learning oracle.

References

- Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- Cutkosky, A. Combining online learning guarantees. In *COLT*, pp. 895–913, 2019.
- Cutkosky, A. and Orabona, F. Black-box reductions for parameter-free online learning in Banach spaces. In *COLT*, pp. 1493–1529, 2018.
- Cutkosky, A. and Sarlos, T. Matrix-free preconditioning in online learning. In *ICML*, pp. 1455–1464, 2019.
- Dekel, O., Flajolet, A., Haghtalab, N., and Jaillet, P. Online learning with a hint. In *NIPS*, pp. 5299–5308, 2017.
- Foster, D. J., Kale, S., Mohri, M., and Sridharan, K. Parameter-free online learning via model selection. In *NIPS*, pp. 6020–6030, 2017.
- Foster, D. J., Rakhlin, A., and Sridharan, K. Online learning: Sufficient statistics and the Burkholder method. In *COLT*, pp. 3028–3064, 2018.
- Hazan, E. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.
- Hazan, E. and Kale, S. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine Learning*, 80(2-3):165–188, 2010.
- Hazan, E. and Megiddo, N. Online learning with prior knowledge. In *COLT*, pp. 499–513, 2007.
- Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- Hazan, E., Rakhlin, A., and Bartlett, P. L. Adaptive online gradient descent. In *NIPS*, pp. 65–72, 2008.
- Jun, K.-S. and Orabona, F. Parameter-free online convex optimization with sub-exponential noise. In *COLT*, pp. 1802–1823, 2019.
- Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *JCSS*, 71(3):291–307, 2005.
- Kempka, M., Kotłowski, W., and Warmuth, M. K. Adaptive scale-invariant online algorithms for learning linear models. In *ICML*, pp. 3321–3330, 2019.
- Kumar, R., Purohit, M., and Svitkina, Z. Improving online algorithms using ML predictions. In *NeurIPS*, pp. 9661–9670, 2018.
- Kumar, R., Purohit, M., Schild, A., Svitkina, Z., and Vee, E. Semi-online bipartite matching. In *ITCS*, pp. 50:1–50:20, 2019.
- Kumar, R., Purohit, M., Svitkina, Z., and Vee, E. Interleaved caching with access graphs. In *SODA*, pp. 1846–1858, 2020.
- Lattanzi, S., Lavastida, T., Moseley, B., and Vassilvitskii, S. Online scheduling via learned weights. In *SODA*, pp. 1859–1877, 2020.
- Li, X. and Orabona, F. On the convergence of stochastic gradient descent with adaptive stepsizes. In *AISTATS*, pp. 983–992, 2019.
- Lykouris, T. and Vassilvitskii, S. Competitive caching with machine learned advice. In *ICML*, pp. 3302–3311, 2018.
- McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *JMLR*, 18(1):3117–3166, 2017.
- McMahan, H. B. and Orabona, F. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *COLT*, pp. 1020–1039, 2014.
- Mohri, M. and Yang, S. Accelerating online convex optimization via adaptive prediction. In *AISTATS*, pp. 848–856, 2016.
- Motwani, R. and Raghavan, P. *Randomized Algorithms*. Cambridge University Press, 1995.
- Orabona, F. and Pál, D. Coin betting and parameter-free online learning. In *NIPS*, pp. 577–585, 2016.
- Pisier, G. *Martingales in Banach Spaces*. Cambridge University Press, 2016.
- Rakhlin, A. and Sridharan, K. Online learning with predictable sequences. In *COLT*, pp. 993–1019, 2013.
- Rohatgi, D. Near-optimal bounds for online caching with machine learned advice. In *SODA*, pp. 1834–1845, 2020.
- Shalev-Shwartz, S. Online learning and online convex optimization. *Foundation and Trends in Machine Learning*, 4(2):107–194, 2011.

Steinhardt, J. and Liang, P. Adaptivity and optimism: An improved exponentiated gradient algorithm. In *ICML*, pp. 1593–1601, 2014.

van der Hoeven, D. User-specified local differential privacy

in unconstrained adaptive online learning. In *NeurIPS*, pp. 14080–14089, 2019.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pp. 928–936, 2003.