RJ10412 (A0708-014) August 24, 2007
Mathematics

IBM Research Report

Online Learning with Prior Knowledge

Elad Hazan, Nimrod Megiddo
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Online Learning with Prior Knowledge

Elad Hazan and Nimrod Megiddo

IBM Almaden Research Center
{hazan,megiddo}@us.ibm.com

Abstract. The standard so-called experts algorithms are methods for utilizing a given set of "experts" to make good choices in a sequential decision-making problem. In the standard setting of experts algorithms, the decision maker chooses repeatedly in the same "state" based on information about how the different experts would have performed if chosen to be followed. In this paper we seek to extend this framework by introducing state information. More precisely, we extend the framework by allowing an experts algorithm to rely on state information, namely, partial information about the cost function, which is revealed to the decision maker before the latter chooses an action. This extension is very natural in prediction problems. For illustration, an experts algorithm, which is supposed to predict whether the next day will be rainy, can be extended to predicting the same given the current temperature.
We introduce new algorithms, which attain optimal performance in the new framework, and apply to more general settings than variants of regression that have been considered in the statistics literature.

1 Introduction

Consider the following standard "experts problem": an online player attempts to predict whether it will rain or not the next day, given advices of various experts. Numerous "experts algorithms" are known, which make predictions based on previous observations and expert advice. These algorithms guarantee that after many iterations the number of mistakes that the algorithm makes is approximately at least as good as that of the best expert in retrospect.

In this paper we address the question of how to utilize prior "state information" in online learning. In the prediction example, suppose that the online predictor has access to various measurements, e.g., temperature and cloud location. Intuitively, this information can potentially improve the performance of the online predictor.

It is not clear a priori how to model prior information in the online learning framework. The information (e.g., temperature) may or may not be correlated with the actual observations (e.g., whether or not it later rains). Even more so, it is conceivable that the state information may be strongly correlated with the observations, but this correlation is very hard to extract. For example, the prior knowledge could be encoded as a solution to a computationally hard problem or even an uncomputable one.

Various previous approaches attempted to learn the correlation between the given information and the observable data. By the above argument, we think such an approach is not robust.

Another approach could be to associate different experts with different decision states and then use standard expert algorithms. The problem here is that the number of states grows exponentially with the dimension of the state space. Therefore, this approach quickly becomes infeasible even for a modest amount of prior information. Other difficulties with this approach arise when the domain of the attributes is infinite.

Perhaps the previous work that is most similar to our approach is the model for portfolio management with side information by Cover and Ordentlich [CO96]. Their approach handles discrete side information, and amounts to handling different side information values as separate problem instances. The measure of performance in their model is standard regret, which must increase in proportion to the available side information.

We propose a framework which does not assume anything about the distribution of the data, prior information, or correlation between the two. The measure of performance is comparative, i.e., based on an extension of the concept of regret. However, unlike [CO96], our model allows the learner to correlate between different states. Our model takes into account the geometric structure of the available information space. As such, it is more similar to the statistical framework of nonparametric regression.

We propose and analyze algorithms which achieve near optimal performance in this framework. Our performance guarantees are valid in both adversarial and stochastic scenarios, and apply to the most general prediction setting, as opposed to previous methods such as nonparametric regression, which apply only to the stochastic scenario and to a more restrictive prediction setting.

We begin with an example of an instance of online learning with state information, in which the information need not be correlated with the observed outcomes. We prove, however, a surprising gain in performance (measured by the standard measure of *regret*) of algorithms that exploit the state information, compared to those that ignore it.

Following this proof of concept, in section 4 we precisely define our model and the measure of performance. We give algorithms and analyze them according to the new performance measure. In section 5 we prove nearly tight lower bounds on the performance of the algorithms, and compare our framework to the well-studied statistical problem of nonparametric regression.

2 Preliminaries

The *online convex optimization* (OCO) problem is defined as follows. The *feasible domain* of the problem is a given convex compact set $\mathcal{P} \subset \mathbb{R}^n$. An adversary picks a sequence of T convex functions $f_t : \mathcal{P} \to \mathbb{R}$, $t = 1, 2, \ldots, T$. The adversary is not restricted otherwise. At time t, $(t = 1, 2, \ldots)$, the decision maker knows only the functions $f_1, \ldots, f_{t-1}$ and has to pick a point $\mathbf{x}_t \in \mathcal{P}$. The decision maker also recalls his previous choices $\mathbf{x}_1 \ldots, \mathbf{x}_{t-1}$. The decision maker is subsequently informed in full about the function f_t, and incurs a cost of $f_t(\mathbf{x}_t)$.

We denote the gradient (resp., Hessian) of a mapping $f : \mathcal{P} \mapsto \mathbb{R}$ at $\mathbf{x} \in \mathcal{P}$ by $\nabla f(\mathbf{x})$ (resp., $\nabla^2 f(\mathbf{x})$). For a family of loss functions $\{f_t(\cdot) : t \in [T]\}$ (henceforth we denote $[n] \triangleq \{1, ..., n\}$) and an underlying convex set $\mathcal{P}$, we denote by $G = \max\{\|\nabla f_t(\mathbf{x})\|_2 : t \in [T],\ \mathbf{x} \in \mathcal{P}\}$ an upper bound on the ℓ_2-norm of the gradients, and by G_∞ an upper bound on the ℓ_∞-norm.

Minimizing Regret. The *regret* of the decision maker after T steps is defined as the difference between the total cost that the decision maker has actually incurred and the minimum cost that the decision maker could have incurred by choosing a certain point repeatedly throughout. More precisely, the regret is equal to

$$R = R(\mathbf{x}_1, \ldots, \mathbf{x}_T\, ;\, f_1, \ldots, f_T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{x}) \ .$$

Denote by $\mathcal{A}$ the *algorithm* that is used by the decision maker to make the sequential choices. Thus, $\mathcal{A}$ is sequence of mappings $(\mathcal{A}_t : t = 1, 2, \ldots)$ so that $\mathbf{x}_t = \mathcal{A}_t(f_1, \ldots, f_{t-1})$. For brevity, denote $\mathbf{x}^T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, $f^T = (f_1, \ldots, f_T)$ and $\mathbf{x}^T = \mathcal{A}(f^{T-1})$. The worst-case regret from an algorithm $\mathcal{A}$ at time T can be defined as

$$\text{Regret}_T(\mathcal{A}) \triangleq \sup_{f^T} R(\mathcal{A}(f^{T-1}), f^T) \ .$$

In other words,

$$\text{Regret}_T(\mathcal{A}) = \sup_{f_1, \ldots, f_T} \left\{ \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right\}$$

The traditional approach to the OCO problem seeks algorithms that minimize the worst-case regret.

Online convex optimization with state information. In this paper we extend the common OCO problem to situations where the decision maker has some information about the "state" prior to his choosing of $\mathbf{x}_t$. We consider specific situations where some state information is revealed to the decision maker. A precise definition is given in section 4.

3 A "proof of concept"

In this section we consider the basic online convex optimization setting over the Euclidean ball $\mathbb{B}_n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \le 1\}$. We assume that each payoff function f_t is linear, i.e., $f_t(\mathbf{x}) = \mathbf{c}_t^\top \mathbf{x}$ for some $\mathbf{c} \in \mathbb{R}^n$ (see see [Zin03] for a reduction from the general OCO problem to the case of linear cost functions). Furthermore, we consider here the case where $\mathbf{c}_t \in [-1, 1]^n$, and assume that only c_{t1}, the first coordinate of $\mathbf{c}_t$, is revealed to the decision maker as state information prior to the choosing of $\mathbf{x}_t \in \mathbb{B}_n$.

The following lower bound is well known for the case where c_{t1} is not revealed to the decision maker prior to choosing $\mathbf{x}_t$ (a similar bound was given in [CBFH$^+$93]; see Appendix for proof):

Lemma 1 (Folk) *For the Online Convex Optimization problem over the Euclidean ball with $\mathbf{c}_t \in [-1,1]^n$ (or even $\mathbf{c}_t \in \{-1,1\}^n$) with no state information, every algorithm has a worst-case regret of at least $\Omega(\sqrt{nT})$.*

We first prove a surprising result that the decision maker can do much better when c_{t1} is known, even if there is no dependency between c_{t1} and the rest of the coordinates of $\mathbf{c}_t$.

Theorem 2 *For the OCO problem over the Euclidean ball with $\mathbf{c}_t \in [-1,1]^n$, in which c_{t1} is bounded away from zero and is revealed to the decision maker as state information prior to the choosing of $\mathbf{x}_t$, there exists an algorithm with a worst-case regret of $O(n^2 \log T)$.*

The condition that c_{t1} is bounded away from zero is intuitively necessary in order to have non-vanishing state information. It is also easy to show that for state information that is identically zero, the lower bound of Lemma 1 holds.

We now analyze the case with prior information, specifically where the decision maker is informed of c_{t1} prior to choosing $\mathbf{x}_t$. The basic idea is to reformulate the OCO problem with state information as an equivalent OCO problem without state information. In our particular case, this can be done by modifying the convex cost function as follows. Suppose the coordinates of $\mathbf{y}_t \equiv (x_{t2}, \ldots, x_{t,n})^\top$ have been fixed so that $\|\mathbf{y}_t\|^2 \le 1$. Then, the optimal choice of x_{t1}, subject to the constraint $\|\mathbf{x}_t\|^2 \le 1$, is

$$x_{t1} = \begin{cases} \sqrt{1 - \|\mathbf{y}_t\|^2} & \text{if } \ c_{t1} < 0 \\ -\sqrt{1 - \|\mathbf{y}_t\|^2} & \text{if } \ c_{t1} \ge 0 \ . \end{cases}$$

In other words,

$$x_{t1} = -\operatorname{sgn}(c_{t1}) \cdot \sqrt{1 - \|\mathbf{y}_t\|^2} \ .$$

It turns out that the cost of choosing $\mathbf{y}_t$ and completing it with an optimal choice of x_{t1} is

$$g_t(\mathbf{y}_t) = c_{t2}x_{t2} + \cdots + c_{tn}x_{tn} - |c_{t1}|\sqrt{1 - \|\mathbf{y}_t\|^2} = \mathbf{u}_t^\top \mathbf{y}_t - |c_{t1}|\sqrt{1 - \|\mathbf{y}_t\|^2} \ ,$$

where

$$\mathbf{u}_t \equiv (c_{t2}, \ldots, c_{tn})^\top \ .$$

Thus, our problem is equivalent to an OCO problem where the decision maker has to choose vector $\mathbf{y}_t$, and the adversary picks cost functions of the form of g_t where c_{t1} is known to the decision maker.

The following algorithm chooses vectors based on weights, which are updated in a multiplicative fashion. Let

$$w_t(\mathbf{y}) = \exp\left\{-\alpha \textstyle\sum_{\tau=1}^{t-1} g_\tau(\mathbf{y})\right\} \ .$$

Thus,

$$w_t(\mathbf{y}) = w_{t-1}(\mathbf{y}) \cdot \exp\left\{-\alpha\, g_{t-1}(\mathbf{y})\right\} \ .$$

The weight function $w_t(\cdot)$ determines the choice of $\mathbf{y}_t$ as follows, where $\mathcal{P} = \mathbb{B}_{n-1}$.

$$\mathbf{y}_t = \frac{\int_{\mathcal{P}} \mathbf{y} \cdot w_t(\mathbf{y})\, d\mathbf{y}}{\int_{\mathcal{P}} w_t(\mathbf{y})\, d\mathbf{y}} \in \mathcal{P} \ .$$

Note that $\mathbf{y}_t$ is a convex combination of points in $\mathcal{P}$ and hence $\mathbf{y}_t \in \mathcal{P}$. The corresponding vector $\mathbf{x}_t$ in the OCO problem with state information is the following:

$$(x_{t2}, \ldots, x_{tn})^\top = \mathbf{y}_t$$

$$x_{t1} = \begin{cases} \sqrt{1 - \|\mathbf{y}_t\|^2} & \text{if } \ c_{t1} < 0 \\ -\sqrt{1 - \|\mathbf{y}_t\|^2} & \text{if } \ c_{t1} \geq 0 \end{cases}$$

We refer to the above-stated algorithm as Alg1. Denote $\rho = \min\{|c_{t1}| \ : \ t = 1, \ldots, T\}$.

Lemma 3 *The worst-case regret of* Alg1 *is at most* $(4n^2/\rho) \cdot \log T$.

The proof of this Lemma is given in the appendix. Briefly, Alg1 belongs to a family of well studied "exponential weighting" algorithms, which can exploit the curvature of the functions $g(\mathbf{y})$, and hence obtain a logarithmic regret. Theorem 2 follows.

Algorithm Alg1 can be implemented in time polynomial in n and T, in a way similar to the implementation of Cover's algorithm [Cov91] by Blum and Kalai [BK97].

4 The general case

To capture state information, we revise the online convex optimization framework as defined in section 2 as follows. We model state information by a vector in a metric space $\mathcal{I}$, which we also call the *information space.* In iteration t, an online algorithm $\mathcal{A}$ accepts besides f^{t-1} and $\mathbf{x}^{t-1}$ also a state vector $\mathbf{k}_t \in \mathcal{I}$ as well as all previous state vector $\mathbf{k}_1, \ldots, \mathbf{k}_{t-1}$.

Henceforth we consider the information space $\mathcal{I}$ as a subset of d-dimensional Euclidean space, even though it makes sense to consider general metric spaces so as to allow the representation of both scalar quantities (e.g., temperature) and problem-specific quanta (e.g., board configurations in the game of chess). The space should at least be metric since the main point of our algorithms is to take advantage of similarity between consecutive state vectors, which is measured according to some distance function. Note that ignoring the similarity between states is equivalent to employing disjoint sets of experts in different states. We also refer to the intrinsic dimensionality of the space, denoted d. For Euclidean space this is the standard dimension, but more generally the notion of *box dimension* [Cla06] is suitable for our applications.

The new performance measure we propose is a strict generalization of the game-theoretic concept of regret as follows.

Definition 1 *For $L > 0$,*

(i) Denote by X_L the family of mappings $\mathbf{x} : \mathcal{I} \mapsto \mathcal{P}$, from the information space to the underlying convex set $\mathcal{P}$, with Lipschitz-constant L, i.e., for all $\mathbf{k}_1, \mathbf{k}_2 \in \mathcal{I}$,

$$\|\mathbf{x}(\mathbf{k}_1) - \mathbf{x}(\mathbf{k}_2)\| \leq L \cdot \|\mathbf{k}_1 - \mathbf{k}_2\| \ .$$

(ii) The L-regret from a sequence of choices $\mathbf{x}_1, ..., \mathbf{x}_T$ is defined as

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X_L} \sum_{t=1}^{T} f_t(\mathbf{x}(\mathbf{k}_t))$$

When L and $\mathbf{k}_1, \ldots, \mathbf{k}_T$ have been fixed, we denote by $\mathbf{x}^*(\cdot)$ a minimizer of $\sum_{t=1}^{T} f_t(\mathbf{x}(\mathbf{k}_t))$ over X_L.

Thus, the actual costs are compared with the costs that could be incurred by the best experts in a family of experts with Lipschitz-constant L. Note that this definition reduces to the standard regret when $L = 0$. If $L = \infty$, then L-regret is the "competitive ratio" studied in competitive analysis of online algorithms. Our model for prior information allows for algorithms which attain sublinear L-regret for $0 < L < \infty$.

4.1 An algorithm for minimizing L-regret

To describe the first algorithm which attains a non-trivial worst-case L-regret, we recall the geometric notion of an ε-net.

Definition 2 *A subset $\mathcal{N} \subseteq \mathcal{I}$ of points in a metric space $\mathcal{I}$ with distance function Δ is called an* ε-net *for the set $S \subseteq \mathcal{I}$ if for every $x \in S$, $\Delta(x, \mathcal{N}) \equiv \inf\{\Delta(x, y) | y \in \mathcal{N}\} \leq \varepsilon$, and in addition $\forall x, y \in \mathcal{N}$. $\Delta(x, y) \geq \varepsilon$.*

The first algorithm, Alg2, which attains a non-trivial worst-case L-regret, constructs an ε-net of the observed data points, denoted $\mathcal{N}$, according to the online greedy algorithm (see [Cla06,KL04]). We also maintain a mapping, denoted $\mathcal{M}$, from all points in $\mathcal{N}$ to the decision space $\mathcal{P}$. Let D denote the diameter of $\mathcal{P}$ and W denote the diameter of the information space $\mathcal{I}$. The algorithm relies on the Lipschitz-constant L and the number of time periods T.

Algorithm Alg2 **(L,T).**

Set $\varepsilon = W \, (D/L)^{2/(d+2)} \; T^{-1/(d+2)}$ and $\mathcal{N} = \emptyset$.

– Given $\mathbf{k}_t \in [0,1]^d$, let $\tilde{\mathbf{k}}_t$ be the state that is closest to $\mathbf{k}_t$ among all state vectors in $\mathcal{N}$, i.e., $\tilde{\mathbf{k}}_t = \arg\min\{\|\mathbf{k} - \mathbf{k}_t\| : \mathbf{k} \in \mathcal{N}\}$.
– Set $\mathbf{x}_t \leftarrow \mathcal{M}(\tilde{\mathbf{k}}_t)$ or, if $t = 0$, then set $\mathbf{x}_0$ arbitrarily.
– Denote by $\prod_{\mathcal{P}}$ the projection operator into the convex set $\mathcal{P}$. Set

$$\mathbf{y} \leftarrow \prod\nolimits_{\mathcal{P}} \left(\mathcal{M}(\tilde{\mathbf{k}}_t) - \tfrac{1}{\sqrt{T}} \nabla f_t(\mathcal{M}(\tilde{\mathbf{k}}_t)) \right)$$

– If $\|\tilde{\mathbf{k}}_t - \mathbf{k}_t\| \leq \varepsilon$, then update $\mathcal{M}(\tilde{\mathbf{k}}_t) \leftarrow \mathbf{y}$ (the size of $\mathcal{N}$ does not increase); else, add $\mathbf{k}_t$ to $\mathcal{N}$ and set $\mathcal{M}(\mathbf{k}_t) \leftarrow \mathbf{y}$.

Theorem 4 *Given L, $\mathcal{P}$, and T,*

$$L\text{-regret}(\textsc{Alg2}) = O\left(W\, G\, L^{1-\frac{2}{d+2}}\, D^{\frac{2}{d+2}} \cdot T^{1-\frac{1}{d+2}} \right)$$

The theorem is proved by independently summing up the L-regret over the "representative" points in the set $\mathcal{N}$. For each such representative point, the optimal strategy in hindsight is almost fixed by diameter considerations. In addition, the total number of such representatives is not too large because the set $\mathcal{N}$ is an ε-net of the observed set of state vectors.

Proof. Summing up the L-regret over the "representative" points in the set $\mathcal{N}$:

$$L\text{-regret}(\textsc{Alg2}) = \sum_{t=1}^{T} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*(\mathbf{k}_t))] = \sum_{\mathbf{k}\in\mathcal{N}} \sum_{t:\tilde{\mathbf{k}}_t = \mathbf{k}} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*(\mathbf{k}_t))] \ .$$

Let $T_{\mathbf{k}} = |\{t \in [T] \mid \tilde{\mathbf{k}}_t = \mathbf{k}\}|$ be the number of iterations during which the prior knowledge $\mathbf{k}_t$ is equal to the representative vector $\mathbf{k} \in \mathcal{N}$. By the properties of the gradient-descent algorithm (Theorem 1 in [Zin03]), for each set of time periods $T_{\mathbf{k}}$, the 0-regret can be bounded as follows.

$$\sum_{t\in T_{\mathbf{k}}} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{P}} \sum_{t\in T_{\mathbf{k}}} f_t(\mathbf{x}) = \sum_{t\in T_{\mathbf{k}}} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*_{\mathbf{k}})] \leq 2GD\sqrt{T_{\mathbf{k}}} \ , \qquad (1)$$

where $\mathbf{x}^*_{\mathbf{k}} = \arg\min \sum_{t\in T_{\mathbf{k}}} f_t(\mathbf{x})$. Also, since for each time period during which $\tilde{\mathbf{k}}_t = \mathbf{k}$ the distance between state vectors is bounded by (using the triangle inequality for the norm),

$$\|\mathbf{x}^*(\mathbf{k}_1) - \mathbf{x}^*(\mathbf{k}_2)\| \leq L \cdot \|\mathbf{k}_1 - \mathbf{k}_2\| \leq L \cdot (\|\mathbf{k}_1 - \mathbf{k}\| + \|\mathbf{k}_2 - \mathbf{k}\|) \leq 2L\varepsilon \ , \quad (2)$$

combining (1) and (2) we get for every $\mathbf{k}$,

$$\begin{aligned}
&\textstyle\sum_{t\in T_{\mathbf{k}}} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*(\mathbf{k}_t))] \\
&\quad = \sum_{t\in T_{\mathbf{k}}} [f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*_{\mathbf{k}})] + \sum_{t\in T_{\mathbf{k}}} [f_t(\mathbf{x}^*_{\mathbf{k}}) - f_t(\mathbf{x}^*(\mathbf{k}_t))] \\
&\quad \leq 2GD\sqrt{T_{\mathbf{k}}} + \sum_{t\in T_{\mathbf{k}}} \nabla f_t(\mathbf{x}^*(\mathbf{k}_t))(\mathbf{x}^*_{\mathbf{k}} - \mathbf{x}^*(\mathbf{k}_t)) \\
&\quad \leq 2GD\sqrt{T_{\mathbf{k}}} + \sum_{t\in T_{\mathbf{k}}} \|\nabla f_t(\mathbf{x}^*(\mathbf{k}_t))\| \cdot \|\mathbf{x}^*(\mathbf{k}_1) - \mathbf{x}^*(\mathbf{k}_t)\| \\
&\quad \leq 2GD\sqrt{T_{\mathbf{k}}} + GT_{\mathbf{k}} \cdot \varepsilon L \ .
\end{aligned}$$

Thus, the total regret is bounded by (using concavity of the square root function)

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}^*(\mathbf{k}_t))] \leq \sum_{\mathbf{k}\in\mathcal{N}}[2GD\sqrt{T_{\mathbf{k}}}+G\varepsilon LT_{\mathbf{k}}] \leq |\mathcal{N}|\cdot 2GD\sqrt{T/|\mathcal{N}|}+G\varepsilon LT\ .$$

It remains to bound the size of $\mathcal{N}$, which is standard for a greedy construction of an ε-net. Since the distance between every two distinct vectors $\mathbf{k}_1, \mathbf{k}_2 \in \mathcal{N}$ is at least ε, by volume arguments and the fact that the information space $\mathcal{I}$ has (box) dimension d, we have $|\mathcal{N}| \leq (W/\varepsilon)^d$. Thus,

$$L\text{-regret}(\textsc{Alg2}) = O\left((W/\varepsilon)^{d/2}\, GD\sqrt{T} + G\varepsilon LT\right)$$

By choosing $\varepsilon = W\,(D/L)^{2/(d+2)}T^{-1/(d+2)}$, we obtain the result.

Remark 1. Algorithm Alg2 receives as input the number of iterations T. This dependence can be removed by the standard "doubling trick" as follows. Apply the algorithm with $t_1 = 100$. Recursively, if the number of iterations exceeds t_{j-1}, then apply Alg2 with $t_j = 2t_{j-1}$ from iteration t_j onwards. The overall regret is

$$\sum_{j=1}^{\log T} WGL^{1-\frac{2}{d+2}}D^{\frac{2}{d+2}}\cdot t_j^{1-\frac{1}{d+2}} \leq \log T\cdot WGL^{1-\frac{2}{d+2}}D^{\frac{2}{d+2}}\cdot T^{1-\frac{1}{d+2}}\ .$$

The same remark shall apply to all consequent variants. For simplicity, we assume henceforth that T is known in advance.

Implementation and running time. It is straightforward to implement Alg2 in time linear in T, n, and d, apart from the projection operator onto the convex set $\mathcal{P}$. This projection is a convex program and can be computed in polynomial time (for various special cases faster algorithms are known).

The performance guarantee of Alg2 decreases exponentially with the dimension of the information space, denoted d. As we show in the next section, this "curse of dimensionality" is inherent in the model, and the bounds are asymptotically tight. Next, we describe an approach to deal with this difficulty.

4.2 Extensions to the basic algorithm

Exploiting low dimensionality of data If the state vectors originate from a lower-dimensional subspace of the information space, the algorithm of the preceding section can be adapted to attain bounds that are proportional to the dimension of the subspace rather than the dimension of the entire information space.

Corollary 5 *Suppose that the prior knowledge vectors $\mathbf{k}_t$ originate from an r-dimensional subspace of $\mathcal{I}$. Then setting $\varepsilon = W(\frac{D}{L})^{2/(r+2)}T^{-1/(r+2)}$ in* Alg2 *we obtain*

$$L\text{-regret}(\textsc{Alg2}) = O(WGL^{1-\frac{2}{r+2}}D^{\frac{2}{r+2}}\cdot T^{1-\frac{1}{r+2}})$$

This corollary follows from the fact that the constructed ε-net in an r-dimensional subspace has size $(W/\varepsilon)^r$ rather than $(W/\varepsilon)^d$.

Specialization to other online convex optimization variants It is possible to modify ALG2 by replacing the online gradient descent step inside the main loop by any other online convex optimization algorithm update. In certain cases this may lead to more efficient algorithms. For example, if the underlying convex set $\mathcal{P}$ is the n-dimensional simplex, then using the ubiquitous Multiplicative-Weights online algorithm (introduced to the learning community by Littlestone and Warmuth [LW94]; see survey [AHK05]) we can obtain the following regret bound

$$L\text{-regret}(\text{MW-ALG2}) = O(WG_\infty L^{1-\frac{2}{d+2}} D^{\frac{2}{d+2}} \cdot T^{1-\frac{1}{d+2}} \sqrt{\log n}) \ .$$

Another possible variant applies a Newton-type update rather than a gradient update. Such second-order algorithms are known to achieve substantial lower regret when the cost functions are exp-convex [HKKA06]. It is also possible to plug in "bandit" algorithms such as [FKM05].

Better ε-nets. The metric embedding literature is rich with sophisticated data structures for constructing ε-nets and computing nearest neighbors over these nets - exactly the geometrical tasks performed by algorithm ALG2. Specifically, it is possible to use the techniques in [KL04] and related papers to obtain algorithms with much better running times.

5 Limitations of learning with prior knowledge

In this section we discuss the limitations of our model for learning with prior knowledge. As a first step, we give lower bounds on the achievable L-regret, which are asymptotically tight up to constant factors.

Following that, we discuss a well-studied statistical methodology, called non-parametric regression, and show that our model generalizes that methodology. As a consequence, the lower bounds proved in the statistics literature apply to our framework and imply lower bounds on the achievable L-regret. These lower bounds are tight in the sense that the algorithms we described in the previous sections attain these bounds up to constant factors.

5.1 Simple lower bounds for L-regret

We begin with a simple lower bound, which shows that the L-regret of any online algorithm with prior information deteriorates exponentially as the dimension grows. Compared to Theorem 4 the bounds are tight up to constant factors.

Lemma 6 *For $\mathcal{P} = [-1, 1]$, $d > 1$, and every $L \geq 0$, the L-regret of any online algorithm is at least $\Omega(GLT^{1-\frac{1}{d}})$.*

Proof. Partition the hypercube $[0, 1]^d$ into $T = \delta^{-d}$ small cubes of edge-length δ. Consider loss functions $f_t(x)$ and prior knowledge vectors $\mathbf{k}_t$ as follows. The sequence of prior knowledge vectors $(\mathbf{k}_1, \ldots, \mathbf{k}_T)$ consists of all centers of the small cubes. Note that for every $i \neq j$, $\|\mathbf{k}_i - \mathbf{k}_j\| \geq \delta$. For each t, independently, pick $f_t = f_t(x)$ to be either Gx or $-Gx$ with equal probability. Note that $\|\nabla f(x)\| = |f'(x)| = G$. Obviously, the expected loss of any algorithm that picks x_t without knowing $f_t(x)$ is zero; thus,

$$\mathbf{E}_{f_1,\ldots,f_t} \left[\sum_{t=1}^{T} f_t(x_t) \right] = 0.$$

Now, define the following function:

$$x^*(\mathbf{k}_t) \triangleq \begin{cases} -\frac{1}{2} L\delta & \text{if } f_t(x) \equiv Gx \\ +\frac{1}{2} L\delta & \text{if } f_t(x) \equiv -Gx \ . \end{cases}$$

The function $x^*(\cdot)$ is in X_L because for every $\mathbf{k}_1$ and $\mathbf{k}_2$,

$$|x^*(\mathbf{k}_1) - x^*(\mathbf{k}_2)| \leq L\delta \leq L \cdot \|\mathbf{k}_1 - \mathbf{k}_2\| \ .$$

Also, the minimum possible total cost using an optimal strategy x^* is

$$\sum_{t=1}^{T} -\tfrac{1}{2} L\delta \cdot G = -T \cdot \tfrac{1}{2} L\delta G = -\tfrac{1}{2} GLT^{1-\frac{1}{d}}$$

where the last equality follows since $T = \delta^{-d}$ and hence $\delta = T^{-\frac{1}{d}}$. Therefore, the expected regret of *any* online algorithm is as claimed.

The previous Lemma does not cover the case of $d = 1$, so for completeness we prove the following lemma.

Lemma 7 *For $d = 1$, prior knowledge space $K = [0, 1]$, $\mathcal{P} = [-1, 1]$, and any $L \geq 0$, the L-regret of any online algorithm is at least $\Omega(G\sqrt{T(\lfloor L \rfloor + 1)})$*

Proof (sketch). Without loss of generality, assume L is an integer. If $L \leq 1$, then this lemma follows from Lemma 1; otherwise, divide the real line $[0, 1]$ into L segments, each of length $\frac{1}{L}$.

The online sequence is as follows. The prior knowledge vectors will be all $L+1$ points $\{k_1, \ldots, k_{L+1}\}$ which divide the segment $[0, 1]$ into L smaller segments. For each such point we have a sequence of $T/(L + 1)$ loss functions $f_t(x)$, each chosen at random, independently, to be either Gx or $-Gx$.

Obviously, the expected payoff of any online algorithm is zero. Now, to define the optimal strategy in hindsight, for each sequence of random functions corresponding to one of the points $\{k_1, \ldots, k_{L+1}\}$, with very high probability, the standard deviation is $O(\sqrt{T/(L + 1)})$. Let $x^*(k_i)$ be either $\frac{1}{4}$ or $-\frac{1}{4}$ according to the direction of the deviation. We claim $x^* \in X_L$ since $|k_1 - k_2| \geq 1/L$ and for all k_1 and k_2,

$$|x^*(k_1) - x^*(k_2)| \leq \frac{1}{2} \leq L \cdot |k_1 - k_2| \ .$$

The loss obtained by x^* is

$$(L + 1) \cdot \frac{1}{4}\sqrt{\frac{T}{L + 1}} = \frac{1}{4}\sqrt{T(L + 1)} \ .$$

This completes the proof.

5.2 The relation to nonparametric regression

Nonparametric regression is the following well-studied problem which can be described as follows. There exists a distribution Ψ on $K \times X$, where that $K \subseteq \mathbb{R}^d$ and $X \subseteq \mathbb{R}$. We are given t samples, $\{(\mathbf{k}_1, x_1), \ldots, (\mathbf{k}_t, x_t)\}$, from this distribution, which we denote by $\mathbf{z}^t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ ($\mathbf{z}_i = (\mathbf{k}_i, x_i)$). The problem is to come up with an estimator for x, given $\mathbf{k} \in \mathbb{R}^d$. An estimator for X which has seen t samples $\mathbf{z}^t$ from the distribution Ψ is denoted by $\theta_t : K \mapsto X$. The goal is to come up with an estimator which is as close as possible to the "optimal" Bayes estimator $\theta(\mathbf{k}) = E[x \,|\, \mathbf{k}]$.

Various distance metrics are considered in the literature for measuring the distance of an estimator from the Bayes estimator. For our purposes it is most convenient to use the L_2-error given by

$$\mathbf{Perf}(\theta_t) \triangleq \mathbf{E}_{(\mathbf{k},x)} \left[(\theta_t(\mathbf{k}) - \theta(\mathbf{k}))^2 \right] \ .$$

The online framework we consider is more general than nonparametric regression in the following sense: an algorithm for online convex optimization with prior information is also an estimator for non-parametric regression, as we show below.

Recall that an algorithm for online optimization $\mathcal{A}$ takes as input the history of cost functions $f_1, \ldots, f_{t-1}$ as well as historical and current state information $\mathbf{k}_1, \ldots, \mathbf{k}_{t-1}, \mathbf{k}_t$, and produces a point in the underlying convex set $x_t = \mathcal{A}(f_1, \ldots, f_{t-1} \ ; \ \mathbf{k}_1, \ldots, \mathbf{k}_t)$. Given an instance of nonparametric regression (K, X), and t samples $\{(\mathbf{k}_i, x_i)\}$, define t cost functions as

$$f_i(x) \triangleq (x - \theta(\mathbf{k}_i))^2 \ .$$

Note that these cost functions are continuous and convex (although not differentiable). Motivated by results on *online-to-batch algorithm conversion*, let the hypothesis of online algorithm $\mathcal{A}$ at iteration t be

$$h_t^{\mathcal{A}}(\mathbf{k}) \triangleq \mathcal{A}(f_1, \ldots, f_{t-1} \ ; \ \mathbf{k}_1, \ldots, \mathbf{k}_{t-1}, \mathbf{k}) \ .$$

Now, define the estimator corresponding to $\mathcal{A}$ by

$$\theta_t^{\mathcal{A}}(\mathbf{k}) \triangleq \tfrac{1}{t} \textstyle\sum_{\tau=1}^{t} h_\tau^{\mathcal{A}}.$$

Standard techniques imply a bound on the performance of this estimator as a function of the L-regret achievable by $\mathcal{A}$:

Lemma 8 *Let L be the Lipschitz constant of the function $\theta : K \mapsto X$. Then,*

$$\lim_{T \mapsto \infty} \Pr_{\mathbf{z}^T \sim \Psi^T} \left[\mathbf{Perf}(\theta_T^{\mathcal{A}}) \leq \frac{1}{T} L\text{-regret}_T(\mathcal{A}) + O\left(\frac{\log T}{\sqrt{T}} \right) \right] = 1 \ .$$

Proof. Standard results of converting online algorithms to batch algorithms, in particular Theorem 2 from [CBCG04], rephrased in our notation, reduces to:

$$\Pr_{\mathbf{z}^t \sim \Psi}\left[\mathbf{E}_{(\mathbf{k},x)\sim\Psi}[f(\theta_t^{\mathcal{A}}(k))] \leq \frac{1}{t}\sum_{\tau=1}^{t-1} f_\tau(h_\tau^{\mathcal{A}}(k_\tau)) + O\left(\frac{1}{\sqrt{t}}\log\frac{1}{\delta}\right)\right] \geq 1-\delta \ .$$

Since for every τ, $f_\tau(\theta(\mathbf{k}_\tau)) = 0$, we obtain

$$\begin{aligned}
1-\delta &\leq \Pr_{\mathbf{z}^t \sim \Psi}\left[\mathbf{E}_{(\mathbf{k},x)\sim\Psi}[f(\theta_t^{\mathcal{A}}(\mathbf{k}))] \leq \frac{1}{t}\sum_{\tau=1}^{t-1} f_\tau(h_\tau^{\mathcal{A}}(\mathbf{k}_\tau)) + O\left(\frac{1}{\sqrt{t}}\log\frac{1}{\delta}\right)\right] \\
&= \Pr_{\mathbf{z}^t \sim \Psi}\left[\mathbf{Perf}(\theta_t^{\mathcal{A}}) \leq \frac{1}{t}\left[\sum_{\tau=1}^{t-1} f_\tau(h_\tau^{\mathcal{A}}(\mathbf{k}_\tau)) - f_\tau(\theta(\mathbf{k}_\tau))\right] + O\left(\frac{1}{\sqrt{t}}\log\frac{1}{\delta}\right)\right] \\
&\leq \Pr_{\mathbf{z}^t \sim \Psi}\left[\mathbf{Perf}(\theta_t^{\mathcal{A}}) \leq \frac{1}{t}\left[L\text{-regret}_T(\mathcal{A})\right] + O\left(\frac{1}{\sqrt{t}}\log\frac{1}{\delta}\right)\right]
\end{aligned}$$

where the equality follows from the definition of $\mathbf{Perf}(\theta_t)$, and in the last inequality we use the fact that $\theta \in X_L$ by our assumption on the Lipschitz constant of θ.

By choosing $\delta = \frac{1}{t}$, with probability approaching 1 we have

$$\mathbf{Perf}(\theta_t^{\mathcal{A}}) \leq \frac{1}{t}\left[L\text{-regret}_T(\mathcal{A})\right] + O\left(\frac{\log t}{\sqrt{t}}\right) \ .$$

Hence, online algorithm with non-trivial L-regret guarantee automatically give a method for producing estimators for nonparameteric regression. In addition, the numerous lower bounds for nonparametric regression that appear in the literature apply to online learning with prior information. In particular, the lower bounds of [Sto82] and [AGK00] show that the exponential dependence of the L-regret is inherent and necessary even for the easier problem of nonparametric regression. It appears that Stone's lower bound [Sto82] has exactly the same asymptotic behavior as achieved in Theorem 4. Closing the gap between the convergence rate $1 - \frac{1}{d+2}$ and our lower bound of $1 - \frac{1}{d}$ is left as an open question.

6 Acknowledgements

We thank Ken Clarkson and Robi Krauthgamer for useful comments and references on ε-nets and nearest neighbor algorithms.

References

[AGK00] A. Antos, L. Györfi, and M. Kohler. Lower bounds on the rate of convergence of nonparametric regression estimates. *Journal of Statistical Planning and Inference*, 83(1):91–100, January 2000.

[AHK05] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. *Manuscript*, 2005.

[BK97] Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. In *COLT '97: Proceedings of the tenth annual conference on Computational learning theory*, pages 309–313, New York, NY, USA, 1997. ACM Press.

[CBCG04] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 2004.

[CBFH$^+$93] Nicolo Cesa-Bianchi, Yoav Freund, David P. Helmbold, David Haussler, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 382–391, New York, NY, USA, 1993. ACM Press.

[CBL06] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, New York, NY, USA, 2006.

[Cla06] Kenneth L. Clarkson. Nearest-neighbor searching and metric space dimensions. In Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006.

[CO96] T.M. Cover and E. Ordentlich. Universal portfolios with side information. 42:348–363, 1996.

[Cov91] T. Cover. Universal portfolios. *Math. Finance*, 1:1–19, 1991.

[FKM05] Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of 16th SODA*, pages 385–394, 2005.

[HKKA06] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT '06: Proceedings of the 19'th annual conference on Computational learning theory*, 2006.

[KL04] R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.

[KW99] Jyrki Kivinen and Manfred K. Warmuth. Averaging expert predictions. In *EuroCOLT '99: Proceedings of the 4th European Conference on Computational Learning Theory*, pages 153–167, London, UK, 1999. Springer-Verlag.

[LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[Sto82] C.J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10:1040–1053, 1982.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference (ICML)*, pages 928–936, 2003.

A Proof of Lemma 1

Proof. Suppose the adversary picks each of the coordinates of $\mathbf{c}_1, \ldots, \mathbf{c}_T$ independently at random from $\{-1, 1\}$. Then, for every algorithm, the expected cost to the decision maker is zero. Given $(\mathbf{c}_1, \ldots, \mathbf{c}_T)$, consider the vector $\mathbf{v} \equiv \sum_{t=1}^{T} \mathbf{c}_t$.

The best vector $\mathbf{x}^* \in \mathbb{B}_n$ with respect to $\mathbf{v}$ is obtained by minimizing $\mathbf{v}^\top \mathbf{x}$ over all $\mathbf{x} \in \mathbb{B}_n$. Obviously, $\mathbf{x}^* = -\mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{v}^\top \mathbf{x}^* = -\mathbf{v}^\top \mathbf{v}/\|\mathbf{v}\| = -\|\mathbf{v}\|$. Thus, the *expected* regret is $\mathbf{E}[\|\mathbf{v}\|]$. By the central limit theorem, each coordinate v_j is distributed approximately as normal with expectation 0 and variance T. It follows that the expected regret is $\mathbf{E}[\|\mathbf{v}\|] = \Omega(\sqrt{nT})$ and hence also the worst-case regret is $\Omega(\sqrt{nT})$.

B Proof of Lemma 3

Proof. Recall that by definition of $g_t(\cdot)$ and the construction of $\mathbf{x}_t$,

$$\mathbf{c}_t^\top \mathbf{x}_t = g_t(\mathbf{y}_t) \ . \tag{3}$$

Let $\mathbf{x}^*$ be the minimizer of $\sum_{t=1}^T \mathbf{c}_t^\top \mathbf{x}$ over $\mathbf{x} \in \mathbb{B}_n$. Recall that

$$\mathbf{v} = \mathbf{c}_1 + \cdots + \mathbf{c}_T$$

and $\mathbf{x}^* = -\mathbf{v}/\|\mathbf{v}\|$. Denote

$$\mathbf{y}^* = (x_2^*, \ldots, x_n^*)^\top \ .$$

It follows that

$$x_1^* = \begin{cases} \sqrt{1 - \|\mathbf{y}^*\|^2} & \text{if } \ v_1 < 0 \\ -\sqrt{1 - \|\mathbf{y}^*\|^2} & \text{if } \ v_1 \geq 0 \end{cases}$$

i.e.,

$$x_1^* = -\operatorname{sgn}(v_1)\sqrt{1 - \|\mathbf{y}^*\|^2} \ .$$

Recall that for every $\mathbf{y}$,

$$g_t(\mathbf{y}) = \sum_{j=1}^n c_{tj} y_j - |c_{t1}|\sqrt{1 - \|\mathbf{y}\|^2} = \mathbf{u}^T \mathbf{y} - |c_{t1}|\sqrt{1 - \|\mathbf{y}\|^2} \ .$$

Therefore, for every t,

$$\mathbf{c}_t^\top \mathbf{x}^* = c_{t1} x_1^* + \mathbf{u}^\top \mathbf{y}^* = -c_{t1} \cdot \operatorname{sgn}(v_1)\sqrt{1 - \|\mathbf{y}^*\|^2} + \mathbf{u}^\top \mathbf{y}^* \geq g_t(\mathbf{y}^*) \ . \tag{4}$$

From (3) and (4) we have

$$\text{Regret}_T(\text{ALG1}) = \sum_{t=1}^T \mathbf{c}_t^\top \mathbf{x}_t - \sum_{t=1}^T \mathbf{c}_t^\top \mathbf{x}^* = \sum_{t=1}^T \mathbf{c}_t^\top \mathbf{x}_t - \mathbf{v}^\top \mathbf{x}^* \leq \sum_{t=1}^T g_t(\mathbf{y}_t) - \sum_{t=1}^T g_t(\mathbf{y}^*)$$

Therefore, we proceed to bound the latter difference. The following notion of convexity called "α-exp-concavity" was introduced by Kivinen and Warmuth [KW99] (see also [CBL06])

Definition 3 (i) *For square matrices of the same order* $\mathbf{P}$ *and* $\mathbf{Q}$*, the notation* $\mathbf{P} \succeq \mathbf{Q}$ *means that* $\mathbf{P} - \mathbf{Q}$ *is positive semidefinite. In other words, for every vector* $\mathbf{x}$*,* $\mathbf{x}^\top \mathbf{P} \mathbf{x} \geq \mathbf{x}^\top \mathbf{Q} \mathbf{x}$.

(*ii*) *For $\alpha > 0$, a twice-differentiable mapping $f : \mathcal{P} \to \mathbb{R}$ is said to be α-*exp-concave *if the mapping $h(\mathbf{x}) \triangleq e^{-\alpha \cdot f(\mathbf{x})}$ is concave.*

Proposition 1 *For $f : \mathbb{R}^n \mapsto \mathbb{R}$, $e : \mathbb{R} \mapsto \mathbb{R}$, and $h = e \circ f$, it holds that $\nabla h(\mathbf{x}) = e'(f(\mathbf{x}))\nabla f(\mathbf{x})$ and hence*

$$\nabla^2 h(\mathbf{x}) = e''(f(\mathbf{x}))\nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^\top + e'(f(\mathbf{x}))\nabla^2 f(\mathbf{x}) \ .$$

The following proposition is proved in [HKKA06].

Proposition 2 *A mapping $f : \mathcal{P} \to \mathbb{R}$ is α-exp-concave if and only if for all $\mathbf{x} \in \mathcal{P}$,*

$$\nabla^2 f(\mathbf{x}) \succeq \alpha \cdot \nabla f(\mathbf{x})(\nabla f(\mathbf{x}))^\top \ .$$

Proposition 3 *The mapping $g_t(\mathbf{y}) = \mathbf{u}_t^\top \mathbf{y} - |c_{t1}|\sqrt{1 - \|\mathbf{y}\|^2}$ is $\frac{\rho}{2n}$-exp-concave.*

Proof. Assume $\rho > 0$ (else the statement is trivially correct). The gradient of g_t is

$$\nabla g_t(\mathbf{y}) = \mathbf{u}_t + \frac{|c_{t1}|}{\sqrt{1 - \|\mathbf{y}\|^2)}}\, \mathbf{y} \ ,$$

hence the Hessian is

$$\nabla^2 g_t(\mathbf{y}) = \frac{|c_{t1}|}{(1 - \|\mathbf{y}\|^2)^{3/2}}\, \mathbf{y}\mathbf{y}^\top + \frac{|c_{t1}|}{\sqrt{1 - \|\mathbf{y}\|^2}}\, \mathbf{I}_{n-1} \ .$$

For the proof we rely on the following relation:

$$(\mathbf{a} + \mathbf{b})(\mathbf{a} + \mathbf{b})^\top \preceq 2(\mathbf{a}\mathbf{a}^\top + \mathbf{b}\mathbf{b}^\top) \ , \qquad (5)$$

which is true because for every vector $\mathbf{w}$,

$$\begin{aligned} \mathbf{w}^\top(\mathbf{a} + \mathbf{b})(\mathbf{a} + \mathbf{b})^\top \mathbf{w} &= (\mathbf{w}^\top \mathbf{a} + \mathbf{w}^\top \mathbf{b})^2 \\ &\le 2[(\mathbf{w}^\top \mathbf{a})^2 + (\mathbf{w}^\top \mathbf{b})^2] = \mathbf{w}^\top[2\mathbf{a}\mathbf{a}^\top + 2\mathbf{b}\mathbf{b}^\top]\mathbf{w} \end{aligned}$$

since $(x + y)^2 \le 2\,(x^2 + y^2)$ for all real x and y. Denoting $\nabla_t = \nabla g_t(\mathbf{y})$, and $\mathbf{u}_t = (c_{t2}, \ldots, c_{tn})^\top$, it follows from (5) that

$$\nabla_t \nabla_t^\top \ \preceq\ 2\mathbf{u}_t\mathbf{u}_t^\top + \frac{2c_{t1}^2}{1 - \|\mathbf{y}\|^2}\, \mathbf{y}\mathbf{y}^\top \ .$$

Since $\|\mathbf{u}_t\|^2 \le n - 1$, it follows that

$$\mathbf{u}_t\mathbf{u}_t^\top \ \preceq\ (n-1)\,\mathbf{I}_{n-1} \ \preceq\ \frac{n-1}{|c_{t1}|}\, \nabla^2 g_t(\mathbf{y}) \ .$$

Also, since $\sqrt{1 - \|\mathbf{y}\|^2} \le 1$ and $|c_{t1}| \le 1$,

$$\frac{c_{t1}^2 \cdot \mathbf{y}\mathbf{y}^\top}{1 - \|\mathbf{y}\|^2} \ \preceq\ \frac{|c_{t1}| \cdot \mathbf{y}\mathbf{y}^\top}{(1 - \|\mathbf{y}\|^2)^{3/2}} \ \preceq\ \nabla^2 g_t(\mathbf{y}) \ .$$

Combining the above relations,

$$\nabla_t \nabla_t^\top \ \preceq\ 2\left(1 + \frac{n-1}{|c_1|}\right) \nabla^2 g_t(\mathbf{y}) \ \preceq\ \frac{2n}{\rho} \nabla^2 g_t(\mathbf{y}) \ .$$

The remainder of the proof follows from the analysis of regret bounds of the EWOO algorithm of [HKKA06], following Blum and Kalai [BK97].