

Online Learning with Low Rank Experts

Elad Hazan

Princeton University

EHAZAN@CS.PRINCETON.EDU

Tomer Koren

Technion—Israel Institute of Technology and Microsoft Research, Herzliya

TOMERK@TECHNION.AC.IL

Roi Livni

The Hebrew University of Jerusalem and Microsoft Research, Herzliya

ROI.LIVNI@MAIL.HUJI.AC.IL

Yishay Mansour

Microsoft Research, Herzliya and Tel Aviv University

MANSOUR@TAU.AC.IL

Abstract

We consider the problem of prediction with expert advice when the losses of the experts have low-dimensional structure: they are restricted to an unknown d -dimensional subspace. We devise algorithms with regret bounds that are independent of the number of experts and depend only on the rank d . For the stochastic model we show a tight bound of $\Theta(\sqrt{dT})$, and extend it to a setting of an approximate d subspace. For the adversarial model we show an upper bound of $O(d\sqrt{T})$ and a lower bound of $\Omega(\sqrt{dT})$.

1. Introduction

Arguably the most well known problem in online learning theory is the so called *prediction with experts advice* problem. In its simplest form, a learner wishes to make an educated decision and at each round chooses to take the advice of one of N experts. The learner then suffers a loss between 0 and 1.

It is a standard result in online learning that, without further assumptions, the best strategy for the learner will incur $\Theta(\sqrt{T \log N})$ regret (Cesa-Bianchi and Lugosi, 2006). However, it is natural to assume that while experts are abundant, their decisions are based on common paradigms and that their decision making is based on few degrees of freedom – for example, if experts are indeed experts, their political bias, social background or school of thought largely dominates their decision making. Experts can also be assets on which the learner wishes to distribute her wealth. In this setting, weather, market condition and interests are dominant factors.

It is also sensible to assume that one can exploit this structure to achieve better regret bounds, potentially independent of the actual number of experts while still maintaining a strategy of picking an expert's advice at each round. Our main result is of this flavor and we show how a learner can exploit hidden structure in the problem in an online setting.

We model the problem as follows: We assume that each expert corresponds to a vector u_i in \mathbb{R}^d space where d is potentially small. Then at each round the experts loss corresponds to a scalar product with a vector v_t chosen arbitrarily, and possibly in an adversarial manner. The learner does not observe the chosen embedding of the experts in Euclidean space nor the vectors v_t , and can only observe the loss of each expert.

To further motivate our setting, let us consider the low rank expert model in the stochastic case. It is well known that for linear predictors in d -dimensional space the regret will be $O(\sqrt{dT})$, independent of the number of experts. Indeed, we show that a simple follow the leader algorithm will achieve this regret bound. In fact, one novelty of this paper is a regret bound that depends on an approximate rank – formally we show that one can improve on the $O(\sqrt{T \log N})$ regret bound and derive bounds that depend on the *approximate rank* rather than the number of experts.

The non-stochastic setting is more challenging. It is true that for linear predictors in d -dimension one can achieve $O(\sqrt{dT})$ regret bound even in the non-stochastic case. But the result assumes that learner has access to the geometric structure of the problem, namely, the embedding of the experts in the Euclidean space. Given the embedding one can apply a *Follow the Regularized Leader* approach with proper regularization to derive the desired regret bound.

Our main result is a regret minimization algorithm that achieves an $O(d\sqrt{T})$ regret in this low d -rank setting, when the learner *does not* have access to the experts’ embedding in Euclidean space. Our algorithm does not need to know the value of the rank d , and adaptively adapts to it. Thus we demonstrate a regret bound that is independent of number of experts. We accompany this upper bound with an $\Omega(\sqrt{dT})$ lower bound.

Our results are part of a larger agenda in online learning. A working premise in Online Learning is that in most cases the stochastic case is the hardest case. Indeed, the literature is filled with generalization bounds and their analogue regret bounds. However, a striking difference is that the statistical bounds are often achieved using simple ERM algorithms, that are oblivious to any structure in the problem, even if the structure is required for the generalization bounds to be valid. In contrast, to achieve the analogue regret bound, one has to work harder. For finite hypothesis class the $\log N$ factor is achieved by a sophisticated algorithm, and for more general convex problems in Euclidean space a problem-specific regularization needs to be invoked in order to achieve optimality. Thus, a key difference is that online algorithms need to be tailored to the structure of the problem. This leads to the disappointing fact that to achieve optimal regret bounds, it is not enough for the problem to be structured but the learner needs to actively understand the structure.

Our current research is an attempt to better understand this key difference: we wish to understand whether an online linear predictor can somehow exploit the geometry of the problem in an implicit manner, similarly to batch ERM algorithms, and how. For this, we invoke a setting where the learner must choose its predictor without the a-priori ability to devise a regularizer. Our findings so far indeed demonstrate that even without access to the structure the learner can indeed overcome her dependence on the irrelevant parameter N .

Technically, one should compare our regret bound of $O(d\sqrt{T})$ to the standard regret bound of $O(\sqrt{T \log N})$. For our bound to be superior one needs that $d = o(\sqrt{\log N})$; while this can indeed be the case in various settings, our result can be better seen as a first step in a more general research direction. We aim to understand how online algorithms can take advantage of structural assumptions in the losses, without being given any explicit information about it.

1.1. Related Work

Low rank assumptions are ubiquitous in the Machine Learning literature. They have been successfully applied to various problems, most notably to matrix completion (Candès and Recht, 2009; Foygel and Srebro, 2011; Srebro et al., 2004) but also in the context of classification with missing

data (Goldberg et al., 2010; Hazan et al., 2015) and large scale optimization (Shalev-Shwartz et al., 2011).

A similar problem that was studied in the literature is the *Branching Experts Problem* (Gofer et al., 2013). In the branching expert problem N potential experts are effectively only k distinct experts, but the clustering of the experts to the k clusters is unknown a-priori. This case can be considered as a special instance of our setting as indeed we can embed each expert as a k -dimensional vector. Gofer et al. (2013) proved a sharp $\Theta(\sqrt{kT})$ regret (the bound is tight only when $k < c \log N$ for some constant $c > 0$). It is perhaps worth noting that when effectively only k experts appear, the stochastic bound is $O(\sqrt{T \log k})$, thus showing that in this similar problem, it is not true that the stochastic case is the hardest case.

Complexity measures for online learning. We are not the first to try and understand what is the proper analogue for ERM in the online setting. Notions like the VC-dimension and Rademacher complexities have been extended to notions of Littlestone-Dimension (Littlestone, 1988; Shalev-Shwartz, 2011), and Sequential Rademacher Complexity (Rakhlin et al., 2010) respectively.

The SOA algorithm suggested by Ben-David et al. (2009) is a general framework for regret minimization that depends solely on the Littlestone dimension. However, the SOA algorithm is conceptually distinct from an ERM algorithm within our framework: to implement the SOA algorithm, one has to have access to the structure of the class (specifically, one needs to compute the Littlestone dimension of subclasses within the algorithm).

Sequential Rademacher complexity seems like a powerful tool for improving our bounds and answering some of our open problems. There are also advances in constructing effective algorithms within this framework (Rakhlin et al., 2012). However, as the branching expert example shows, there is no general argument that show that structure in the problem leads to stochastic-analogue bounds on the complexity.

Learning from easy data. In another line of research, which is similar in spirit to ours, several authors attempt to go beyond worst-case analysis in online learning, and provide algorithms and bounds that can exploit deficiencies in the data. Work in this direction includes the study of worst-case robust online algorithms that can also adapt to stochastic i.i.d. data (e.g., Hazan and Kale, 2009; Rakhlin et al., 2013; De Rooij et al., 2014; Sani et al., 2014), as well as the exploration of various structural assumptions that can be leveraged for obtaining improved regret guarantees (e.g., Cesa-Bianchi et al., 2007; Hazan and Kale, 2010, 2011; Chiang et al., 2012; Rakhlin and Sridharan, 2013). However, to the best of our knowledge, low rank assumptions in online learning have not been explored in this context.

Adaptive online algorithms. Online adaptive learning methods have recently been the topic of extensive study and are effective for large scale stochastic optimization in practice. One of the earliest and most widely used methods in this family is the AdaGrad algorithm (Duchi et al., 2011), a subgradient descent method that dynamically incorporate knowledge of the geometry of the data from earlier iterations. Our problem can be cast into an online linear optimization problem and subgradient descent methods are indeed applicable. It might seem at first sight that adapting the regularization via AdaGrad can lead to desired results. However, the analysis of the AdaGrad algorithm can only yield an $O(\sqrt{dNT})$ bound on the regret in our low-rank setting. In fact, a closer inspection reveals that the \sqrt{N} factor in the latter bound is unavoidable for AdaGrad: as we show in Appendix B, in our setting the regret of AdaGrad is lower bounded by $\Omega(\min\{\sqrt{N}, T\})$.

2. Problem Setup and Main Results

We recall the standard adversarial online experts model for T rounds with N experts. At each round $t = 1, \dots, T$, the learner chooses a probability vector $x_t \in \Delta_N$, where Δ_N denotes the N -simplex, namely the set of all possible distributions over N experts,

$$\Delta_N = \left\{ x \in \mathbb{R}^N : \forall i, x(i) \geq 0 \text{ and } \sum_{i=1}^N x(i) = 1 \right\}.$$

An adversary replies by choosing a loss vector $\ell_t \in [-1, 1]^N$,¹ and the learner suffers a loss $x_t(\ell_t) = x_t \cdot \ell_t$. The objective of the learner is to minimize her regret, which is defined as follows,

$$\text{Regret}_T = \sum_{t=1}^T x_t \cdot \ell_t - \min_{i \in [N]} \sum_{t=1}^T \ell_t(i).$$

In the stochastic online experts model, the adversary selects a distribution \mathcal{D} over the loss vectors in $[-1, 1]^N$, and at time t a random $\ell_t \in [-1, 1]^N$ is selected from \mathcal{D} . The regret is.

$$\text{Regret}_T = \sum_{t=1}^T x_t \cdot \mathbb{E}[\ell_t] - \min_{i \in [N]} \sum_{t=1}^T \mathbb{E}[\ell_t(i)],$$

where the expectations are taken over the random loss vectors selected from \mathcal{D} .

In our setting, we wish to assume that there is a structure over the experts which implies that the loss vectors are structured, and are derived from a low rank subspace. Therefore we will add the following constraint over the adversary: let $L \in \mathbb{R}^{N \times T}$ be the loss matrix obtained in hindsight (i.e., the t 'th column of L is ℓ_t). We restrict the feasible strategies for the adversary to only such that satisfy:

$$\text{rank}(L) = d.$$

An equivalent formulation of our model is as follows: An adversary chooses at the beginning of the game a matrix $U \in \mathbb{R}^{N \times d}$, where each row corresponds to an expert. At round t the adversary chooses a vector v_t , and the learner gets to observe ℓ_t where $\ell_t = Uv_t$. The objective of the learner remains the same: to choose at each round a probability distribution x_t that minimizes the regret. We stress that the learner observes only the loss vectors ℓ_t , and does not have access to either U or the vectors v_t .

2.1. Main Results

We next state the main results of this paper:

Theorem 1 *The T -round regret of Algorithm 2 (described in Section 4 below) is at most $O(d\sqrt{T})$, where $d = \text{rank}(L)$.*

We remark that a regret upper bound of $O(\sqrt{T} \min\{d, \sqrt{\log N}\})$ is attainable by combining the standard multiplicative-updates algorithm with our algorithm.² Our upper bound is accompanied by the following lower bound.

-
1. As will become apparent later, in our setup it is more natural to consider symmetric $[-1, 1]$ loss values rather than the typical $[0, 1]$ losses. The two variants of the problem are equivalent up to a simple shift and scaling of the losses—a transformation that preserves the rank of the loss matrix.
 2. A standard way to accomplish this is by running the two online algorithms in parallel, and choosing between their predictions by treating them as two meta-experts in another multiplicative-weights algorithm.

Theorem 2 *For any online learning algorithm, T and $d \leq \log_2 N$, there exists a sequence of loss vectors $\ell_1, \dots, \ell_T \in [-1, 1]^N$ such that*

$$\text{Regret}_T = \sum_{t=1}^T x_t \cdot \ell_t - \min_{i \in [N]} \sum_{t=1}^T \ell_t(i) \geq \sqrt{\frac{dT}{8}},$$

and $\text{rank}(L) = d$.

3. Preliminaries

3.1. Notation

Let I_n be the $n \times n$ identity matrix. Let $\mathbf{1}_n$ be a vector of length n with all 1 entries. The columns of a matrix U are denoted by u_1, u_2, \dots . The i 'th coordinate of a vector x is denoted by $x(i)$. For a matrix M , we denote by M^\dagger the Moore-Penrose pseudo-inverse of M . For a positive definite matrix $H > 0$ we will denote its corresponding norm $\|x\|_H = \sqrt{x^\top H x}$, and its dual norm $\|x\|_H^* = \sqrt{x^\top H^{-1} x}$. Given a positive semi-definite matrix $M \geq 0$ its corresponding Ellipsoid is defined as:

$$\mathcal{E}(M) = \{x : x^\top M^\dagger x \leq 1\}.$$

3.2. Ellipsoidal Approximation of Convex Bodies

A main tool in our algorithm is an Ellipsoid approximation of convex bodies. Recall John's theorem for symmetric zero-centered convex bodies.

Theorem 3 (John's Theorem; e.g, Ball, 1997) *Let K be a convex body in \mathbb{R}^d that is symmetric around zero (i.e., $K = -K$). Let \mathcal{E} be an ellipsoid with minimum volume enclosing K . Then:*

$$\frac{1}{\sqrt{d}} \mathcal{E} \subseteq K \subseteq \mathcal{E}.$$

While computing the minimum volume enclosed ellipsoid is computationally hard, for symmetric convex bodies it can be approximated to within $1 + \epsilon$ factor in polynomial time. Specifically, given as input a matrix $A \in \mathbb{R}^{N \times d}$, consider the polytope $P_A = \{x : \|Ax\|_\infty \leq 1\}$. We have the following.

Theorem 4 (Grötschel et al., 2012, Theorem 4.6.5) *There exists a poly-time procedure $\text{MVVE}(A)$ that receives as input a matrix $A \in \mathbb{R}^{N \times d}$ and returns a matrix M such that*

$$\frac{1}{\sqrt{2d}} \mathcal{E}(M) \subseteq P_A \subseteq \mathcal{E}(M).$$

3.3. Online Mirror Descent

Another main tool in our analysis is the well-known *Online Mirror Descent* (OMD) algorithm for online convex optimization. The Online mirror descent is a subgradient descent method for optimization over a convex set in \mathbb{R}^d that employs a regularization term, chosen a-priori. In Algorithm 1 we describe the algorithm for the special case where the convex set is Δ_N and the regularization function is chosen to be $\|\cdot\|_H^2$ for some input matrix $H > 0$:

The regret bound of the algorithm is dependent on the choice of regularization and is given as follows:

Algorithm 1 Online Mirror Descent (OMD)

-
- 1: **input:** $H > 0$, $\{\eta_t\}_{t=1}^T$, $x_1 \in \Delta_N$.
 - 2: **for** $t = 1$ to T **do**
 - 3: Play x_t
 - 4: Suffer cost $x_t \cdot \ell_t$ and observe ℓ_t
 - 5: Update

$$x_{t+1} = \arg \min_{x \in \Delta_N} \ell_t \cdot x + \eta_t^{-1} \|x - x_t\|_H^2.$$

- 6: **end for**
-

Lemma 5 (e.g., Hazan, 2015) *The T -round regret of the OMD algorithm (Algorithm 1) is bounded as follows:*

$$\sum_{t=1}^T \ell_t \cdot x_t - \sum_{t=1}^T \ell_t \cdot x^* \leq \frac{1}{\eta_T} \|x_1 - x^*\|_H^2 + \frac{1}{2} \sum_{t=1}^T \eta_t (\|\ell_t\|_H^*)^2.$$

3.4. Rademacher Complexity

Our tool to analyze the stochastic case will be the Rademacher Complexity, specifically we will use it to bound the regret of a “Follow The Leader” algorithm (FTL). Recall that the FTL algorithm selection rule is defined as follows:

$$x_t = \arg \min_{x \in \Delta_N} \sum_{i=1}^{t-1} \ell_i \cdot x.$$

One way to bound the regret of the FTL algorithm in the stochastic case is by bounding the Rademacher complexity of the feasible samples. Recall that the Rademacher Complexity of a class of target function \mathcal{F} over a sample $S_t = \{\ell_1, \dots, \ell_t\}$ is defined as follows

$$R(\mathcal{F}, S_t) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{t} \sum_{i=1}^t \sigma_i f(\ell_i) \right],$$

where $\sigma \in \{-1, 1\}^t$ are i.i.d. Rademacher distributed random variables. The following bound is standard and well known, and for completeness we provide a proof in Appendix A.1.³

Lemma 6 *Let K be a symmetric convex set centered around zero in \mathbb{R}^d . Recall that the dual set K^* is defined as follows:*

$$K^* = \{x : \sup_{y \in K} |y \cdot x| \leq 1\}.$$

Let $S_t = \{\ell_1, \dots, \ell_t\} \subseteq K$ and let $\mathcal{F} \subseteq \alpha K^*$ be a subclass of linear functions, then:

$$R(\mathcal{F}, S_t) \leq \alpha \sqrt{\frac{d}{t}}.$$

Another standard bound applies to the case where \mathcal{F} is bounded in the l_1 -norm.

3. Surprisingly, we could not find any specific reference that precisely derives it.

Lemma 7 (Kakade et al., 2009) *Let $S_t = \{\hat{\ell}_1, \dots, \hat{\ell}_t\} \in \mathbb{R}^N$ and let \mathcal{F}_1 be a subclass of linear functions such that $\sup\{\|f\|_1 : f \in \mathcal{F}\} \leq 1$, then:*

$$R(\mathcal{F}_1, S_t) \leq \max_i \|\hat{\ell}_i\|_\infty \sqrt{\frac{2 \log N}{t}}.$$

The Rademacher complexity is a powerful tool in statistical learning theory and it allows us to bound the generalization error of an FTL algorithm. Namely, for every sample $S_t = \{\ell_1, \dots, \ell_t\}$ denote:

$$f_S = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^t f(\ell_i).$$

Then we have the following bound for every $f^* \in \mathcal{F}$ (for i.i.d. loss vectors; see for example [Shalev-Shwartz and Ben-David, 2014](#)):

$$\mathbb{E}_{S_t \sim D} \mathbb{E}_{\ell \sim D} [f_{S_t}(\ell) - f^*(\ell)] \leq 2 \mathbb{E}_{S_t \sim D} [R(\mathcal{F}, S_t)].$$

Applying this to FTL in the experts setting we have, in terms of regret, that for any x^* :

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T \ell_t \cdot x_t - \ell_t \cdot x^* \right] &= \sum_{t=1}^T \mathbb{E}_{\ell_1, \dots, \ell_{t-1} \sim D} \mathbb{E}_{\ell_t \sim D} [\ell_t \cdot x_t - \ell_t \cdot x^*] \\ &\leq 2 \sum_{t=1}^T \mathbb{E}_{S_{t-1} \sim D} [R(\Delta_N, S_{t-1})]. \end{aligned} \tag{1}$$

4. Upper Bound

In this section we discuss our online algorithm for the adversarial model, which is given in [Algorithm 2](#). The algorithm is a version of Online Mirror Descent with adaptive regularization. It maintains a positive-definite matrix H , which is being updated whenever the newly observed loss vector ℓ_t is not in the span of previously appeared losses. In all other time steps—i.e., when ℓ_t remains in the previous span—the algorithm performs an Online Mirror Descent type update (see [Algorithm 1](#)), with the function $\|x\|_H^2 = x^\top H x$ as a regularizer.

The algorithm updates the regularization matrix H so as to adapt to the low-dimensional geometry of the set of feasible loss vectors. Indeed, as our analysis below reveals, H is an ellipsoidal approximation of a certain low-dimensional convex set in \mathbb{R}^N to which the loss vectors ℓ_t can be localized. This low-dimensional set is the intersection of the unit cube in N dimensions—in which the loss vectors ℓ_t reside by definition—and the low dimensional subspace spanned by previously observed loss vectors, given by $\text{span}(U)$. Whenever the latter subspace changes, namely, once a newly observed loss vector leaves the span of previous vectors, the ellipsoidal approximation is recomputed and the matrix H is updated accordingly.

To derive [Theorem 1](#), we begin with analyzing a simpler case where the learner is aware of the subspace from which losses are derived. Specifically, assume that at the beginning of the rounds, the learner is equipped with a rank d matrix U such that for all losses $\ell_1, \ell_2, \dots \in \text{span}(U)$ where we denote by $\text{span}(U)$ the span of the columns of the matrix U .

Algorithm 2 Online Low Rank Experts

- 1: Initialize: $x_1 = \frac{1}{N}\mathbf{1}_N$, $\tau = 0$, $k = 0$, $U = \{\}$
- 2: **for** $t = 1$ to T **do**
- 3: Observe ℓ_t , suffer cost $x_t \cdot \ell_t$.
- 4: **if** $\ell_t \notin \text{span}(U)$ **then**
- 5: Add ℓ_t as a new column of U , reset $\tau = 0$, and set $k \leftarrow k + 1$.
- 6: Compute $M = \text{MVEE}(U^\top)$ and $H = I_n + U^\top MU$.
- 7: **end if**
- 8: let $\tau \leftarrow \tau + 1$ and $\eta_t = 4\sqrt{k/\tau}$, and set:

$$x_{t+1} = \arg \min_{x \in \Delta_N} \ell_t \cdot x + \eta_t^{-1} \|x - x_t\|_H^2.$$

- 9: **end for**
-

In this simplified setting, we can obtain a regret bound of $O(\sqrt{dT})$ via John's theorem (Theorem 3).⁴ As discussed above, the loss vectors ℓ_1, \dots, ℓ_T can be localized to the intersection of the unit cube in N dimensions with the d -dimensional subspace spanned by the columns of U . Then, John's theorem asserts that the minimal-volume enclosing ellipsoid of the intersection is a \sqrt{d} -approximation to the set of feasible loss vectors.

Theorem 8 *Run Algorithm 1 with Input H , $\{\eta_t\}$ and x_1 defined as follows: (i) $H = I_n + U^\top MU$, where $M = \text{MVEE}(U^\top)$, (ii) $\eta_t = 4\sqrt{d/t}$, where $d = \text{rank}(U)$, and (iii) $x_1 \in \Delta$ is arbitrary. If $\ell_1, \dots, \ell_T \in \text{span}(U)$, then the expected T -round regret of the algorithm is at most $8\sqrt{dT}$.*

Proof Consider the d -dimensional polytope

$$P = \{v \in \mathbb{R}^d : \|U^\top v\|_\infty \leq 1\}.$$

Then by John's Theorem (Theorem 3), we have,

$$\mathcal{E}\left(\frac{1}{2d}M\right) \subseteq P \subseteq \mathcal{E}(M). \quad (2)$$

In order to apply Lemma 5, we need to bound both $\|\ell_t\|_H^*$ and $\|x_1 - x^*\|_H^2$. We first bound the norms $\|\ell_t\|_H^*$. Notice that for each loss vector ℓ_t there exists $v_t \in P$ such that $\ell_t = U^\top v_t$ (as $\ell_t \in \text{span}(U)$ and $\|\ell_t\|_\infty \leq 1$). Thus, we can write,

$$(\|\ell_t\|_H^*)^2 = \ell_t^\top H^{-1} \ell_t = v_t^\top U (I_n + U^\top MU)^{-1} U^\top v_t \leq v_t^\top U (U^\top MU)^\dagger U^\top v_t = v_t^\top M^{-1} v_t,$$

where we have used the fact that $U(U^\top MU)^\dagger U^\top = M^{-1}$ for matrices U and $M > 0$ (see Lemma 11 in Appendix A). Now, since $v_t \in P$ and $\mathcal{E}(M)$ is enclosing P , we obtain $v_t^\top M^{-1} v_t \leq 1$. This proves that $(\|\ell_t\|_H^*)^2 \leq 1$.

Next we bound $\|x_1 - x^*\|_H \leq 2$. Since $\|x_1 - x^*\|_H \leq 2 \max_{x \in \Delta_n} \|x\|_H$, it suffices to bound $\max_{x \in \Delta_n} \|x\|_H$. Hence, our goal is to show that $\|x\|_H \leq 2\sqrt{d}$ for all $x \in \Delta_n$. Since $\|x\|_H^2 =$

4. We remark that for the simplified setting, the $O(\sqrt{dT})$ regret bound is in fact tight, as our $\Omega(\sqrt{dT})$ lower bound (given in Section 5) applies in a setting where the subspace of the loss vectors is known a-priori to the learner.

$1 + 2d \|x\|_{H'}^2$ with $H' = \frac{1}{2d}U^\top MU$, it is enough to bound the norm $\|x\|_{H'}^2$. Given a convex set in \mathbb{R}^d , recall that the dual set is given by

$$P^* = \{x : \sup_{p \in P} |x \cdot p| \leq 1\}.$$

The dual of an ellipsoid $\mathcal{E}(M)$ is given by $(\mathcal{E}(M))^* = \mathcal{E}(M^{-1})$ and it is standard to show that Eq. (2) implies in the dual:

$$(\mathcal{E}(M))^* \subseteq P^* \subseteq (\mathcal{E}(\frac{1}{2d}M))^*.$$

Taken together we obtain that $P^* \subseteq \mathcal{E}(2dM^{-1})$. Note that by definition the columns of U are in P^* , hence, for every u_i , $\|u_i\|_M^2 \leq 2d$. Since $x \in \Delta_N$,

$$\|x\|_{H'}^2 = \frac{1}{2d}\|Ux\|_M^2 \leq \frac{1}{2d} \max_i \|u_i\|_M^2 \leq 1.$$

Equipped with the bounds $\|x\|_H \leq \sqrt{1+2d} \leq 2\sqrt{d}$ for all $x \in \Delta_n$ and $\|\ell_t\|_H^* \leq 1$ for all t , we are now ready to analyze the regret of the algorithm, which via Lemma 5 can be bounded as follows:

$$\begin{aligned} \text{Regret}_T &= \sum_{t=1}^T \ell_t \cdot x_t - \sum_{t=1}^T \ell_t \cdot x^* \\ &\leq \frac{1}{\eta_T} \|x_1 - x^*\|_H^2 + \frac{1}{2} \sum_{t=1}^T \eta_t (\|\ell_t\|_H^*)^2 && \because \text{Lemma 5} \\ &\leq \frac{4}{\eta_T} \max_{x \in \Delta_n} \|x\|_H^2 + \frac{1}{2} \sum_{t=1}^T \eta_t (\|\ell_t\|_H^*)^2 && \because \|x_1 - x^*\|_H \leq 4 \max_{x \in \Delta_n} \|x\|_H \\ &\leq \frac{16d}{\eta_T} + \frac{1}{2} \sum_{t=1}^T \eta_t && \because \max_{x \in \Delta_n} \|x\|_H^2 \leq 4d, \|\ell_t\|_H^* \leq 1. \end{aligned}$$

A choice of $\eta_t = 4\sqrt{d/t}$, together with the inequality $\sum_{t=1}^T 1/\sqrt{t} \leq 2\sqrt{T}$, gives the theorem. \blacksquare

The d -low rank setting does not assume that the learner has access to the subspace U , and potentially an adversary may adapt her choice of subspace to the learner's strategy. However, the learner can still obtain regret bounds that are independent of the number of experts. We are now ready to prove Theorem 1.

Proof of Theorem 1 Let $t_0 = 1$, $t_{d+1} = T$ and for all $1 \leq k \leq d$ let t_k be the round where the k 'th column is added to U . Also, let $T_k = t_{k+1} - t_k$ the length of the k 'th epoch. Notice that between rounds t_k and t_{k+1} the algorithm's execution is identical to Algorithm 1 with input depicted in Theorem 8. Therefore its regret in this time period is at most $8\sqrt{kT_k}$. The total regret is then bounded by

$$8 \sum_{k=0}^d \sqrt{kT_k} \leq 8 \sqrt{\sum_{k=0}^d k} \cdot \sqrt{\sum_{k=0}^d T_k} \leq 8d\sqrt{T},$$

and the theorem follows. \blacksquare

4.1. Stochastic Online Experts

We now turn to analyze the regret in the stochastic model, where the loss vectors ℓ_t are chosen i.i.d. from some unknown distribution. In this case we can achieve a right regret bound of $O(\sqrt{dT})$ using a simple ‘‘Follow The Leader’’ (FTL) algorithm. We will in fact show an even stronger result for the stochastic case, that an approximate rank is enough to bound the complexity. Recall that the approximate rank, $\text{rank}_\epsilon(L)$, of a matrix is defined as follows (see [Alon et al., 2013](#)):

$$\text{rank}_\epsilon(L) = \min\{\text{rank}(L') : \|L' - L\|_\infty < \epsilon\}.$$

The following statement is the main result for this section:

Theorem 9 *Assume that an adversary chooses her losses $\{\ell_t\}$ i.i.d. from some distribution \mathcal{D} supported on $[-1, 1]^N$. Then the T -round regret of the FTL algorithm is bounded by:*

$$\text{Regret}_T \leq 8\mathbb{E} \left[\sqrt{T \cdot \text{rank}_\epsilon(L)} \right] + \epsilon\sqrt{T \log N},$$

for every $0 \leq \epsilon < 1$. In particular, if $\text{rank}(L) \leq d$ almost surely, then $\text{Regret}_T = O(\sqrt{dT})$.

Proof Our proof relies on Eq. (1) and a bound for $R(\Delta_N, S_t)$. Fix a sequence $S_T = \{\ell_1, \dots, \ell_T\}$ and let $d = \text{rank}_\epsilon(L)$ and let U be $N \times d$ matrix such that

$$L = UV + \hat{L},$$

where $\max_{i,j} |\hat{L}_{i,j}| < \epsilon$. We will denote the columns of \hat{L} by $\hat{\ell}_1, \dots, \hat{\ell}_N$. We define a symmetric convex set centered around zero in \mathbb{R}^d as follows:

$$K = \{v : \sup_i |u_i \cdot v| \leq 2\}.$$

Note that for every v_t we have that $v_t \in K$ if $\epsilon \leq 1$. By definition of the set we have: $u_i \in 2K^*$ for every i . One can verify that K^* is convex, hence if we let $\mathcal{F} = \text{conv}(u_1, \dots, u_N)$ we have that $\mathcal{F} \subseteq 2K^*$. We can think of \mathcal{F} as a linear function space, where $f_u(v) = u \cdot v$. It follows by Lemma 6 that $R(\mathcal{F}, S_t) \leq \sqrt{2d/t}$. Finally,

$$R(\Delta_N, S_t) = \mathbb{E} \left[\sup_{x \in \Delta_N} \sum_{i=1}^t \frac{1}{t} \sigma_i x \cdot \ell_i \right] \leq \mathbb{E} \left[\sup_{x \in \Delta_N} \frac{1}{t} \sum_{i=1}^t \sigma_i x \cdot Uv_i \right] + \mathbb{E} \left[\sup_{x \in \Delta_N} \frac{1}{t} \sum_{i=1}^t \sigma_i x \cdot \hat{\ell} \right].$$

Next, we have:

$$\begin{aligned} \mathbb{E} \left[\sup_{x \in \Delta_N} \frac{1}{t} \sum_{i=1}^t \sigma_i x \cdot Uv_i \right] &= \mathbb{E} \left[\sup_{x \in \Delta_N} \frac{1}{t} \sum_{i=1}^t \sigma_i (U^\top x) \cdot v_i \right] \\ &= \sup_{f \in \text{conv}(u_i)} \mathbb{E} \left[\frac{1}{t} \sum_{i=1}^t \sigma_i f(v_i) \right] \\ &= R(\mathcal{F}, S_t) < 2\sqrt{\frac{d}{t}}. \end{aligned} \tag{3}$$

and by Lemma 7,

$$\mathbb{E} \sup_{x \in \Delta_N} \left[\frac{1}{t} \sum_{i=1}^t \sigma_i x \cdot \hat{\ell}_i \right] \leq \epsilon \sqrt{\frac{2 \log N}{t}}. \tag{4}$$

Taking Eq. (3) and Eq. (4) together, we have:

$$R(\Delta_N, S_t) \leq 2\sqrt{\frac{\text{rank}_\epsilon L}{t}} + \epsilon\sqrt{\frac{2 \log N}{t}}.$$

The statement now follows from Eq. (1). ■

5. Lower Bound

We now prove Theorem 2. For our proof we will rely on lower bounds for online learning of hypotheses classes with respect to the Littlestone dimension (see [Shalev-Shwartz, 2011](#)). For a class \mathcal{H} of target functions $h : \mathcal{X} \rightarrow \{0, 1\}$, the Littlestone dimension $\text{Ldim}(\mathcal{H})$ measures the complexity, or online learnability, of the class.

To define $\text{Ldim}(\mathcal{H})$ one considers trees whose internal nodes are labeled by instances. Any branch in such a tree can be described as a sequence of examples $(x_1, y_1), \dots, (x_d, y_d)$ where x_i is the instance associated with the i th node in the path, and y_i is 1 if x_{i+1} is the right child of the i -th node, and $y_i = 0$ otherwise. $\text{Ldim}(\mathcal{H})$ is then defined as the depth of the largest binary tree that is shattered by \mathcal{H} . An instance-labeled tree is said to be shattered by a class \mathcal{H} if for any root-to-leaf path $(x_1, y_1), \dots, (x_d, y_d)$ there is some $h \in \mathcal{H}$ such that $h(x_i) = y_i$.

To prove Theorem 2, we need the following result about the Littlestone dimension.

Lemma 10 ([Ben-David et al., 2009](#)) *Let \mathcal{H} be any hypothesis class with finite $\text{Ldim}(\mathcal{H})$, where Ldim is the Littlestone-dimension of a class \mathcal{H} . For any (possibly randomized) algorithm, there exists a sequence of labeled instances $(v_1, y_1), \dots, (v_T, y_T)$ with $y_t \in \{0, 1\}$ such that*

$$\mathbb{E} \left[\sum_{t=1}^T |\hat{y}_t - y_t| \right] - \min_{h \in \mathcal{H}} \sum_{i=1}^T |h(x_t) - y_t| \geq \sqrt{\frac{\text{Ldim}(\mathcal{H})T}{8}}.$$

where \hat{y}_t is the algorithm's output at iteration t .

Proof of Theorem 2 We let \mathcal{H} be the 2^d vertices of the d -dimensional hypercube. We define a function class \mathcal{F} over the domain $\mathcal{X} = \{e_1, \dots, e_d\}$ of standard basis vectors. A function $f_u \in \mathcal{F}$, is labeled by $u \in \mathcal{H}$, and defined over the set of basis vector e_j , as follows,

$$f_u(e_j) = \begin{cases} 0 & \text{if } u(j) = -1, \\ 1 & \text{if } u(j) = 1. \end{cases}$$

One can verify that $\text{Ldim}(\mathcal{F}) = d$. For each $u_i \in \mathcal{H}$ and $y \in \{0, 1\}$, we can write

$$|f_{u_i}(e_j) - y| = \frac{1 - (2y - 1) \cdot u_i \cdot e_j}{2}.$$

By Lemma 10, we deduce that for any algorithm, there exists a sequence $(v_1, \bar{y}_1), \dots, (v_T, \bar{y}_T)$ of standard basis vectors v_1, \dots, v_T and $\bar{y}_1, \dots, \bar{y}_T \in \{-1, 1\}$ such that:

$$\sum_{t=1}^T \sum_i x_t(i) u_i \cdot (\bar{y}_t v_t) - \min_u \sum_{i=1}^T u \cdot (\bar{y}_t v_t) \geq 2\sqrt{\frac{dT}{8}}. \quad (5)$$

We now consider an adversary that chooses U as his expert matrix, and at round t the learner observes $\ell_t = U(\bar{y}_t v_t)$. The lower bound now follows from Eq. (5); the fact that $\text{rank}(L) = d$ follows from the fact that our experts are embedded in \mathbb{R}^d . ■

6. Discussion and Open Problems

We considered the problem of experts with a hidden low rank structure. Our findings are that in the non-stochastic case, similar to the stochastic case, the regret bounds are independent of the number of experts. The most natural question is then to bridge the gap between the upper and lower bounds:

Open Problem 1 *Is there an algorithm that can achieve regret $O(\sqrt{dT})$ for any sequence ℓ_1, \dots, ℓ_T such that $\text{rank}(L) = d$? Alternatively, can one prove a lower bound of $\Omega(d\sqrt{T})$?*

As discussed, our agenda is more general than the low-rank setting. Our aim is to construct new online algorithms that can exploit structure in the data, without explicit information on the structure. Other settings can also be considered within our framework.

Another interesting setting, that avoids dependence in dimension, is to assume that experts are embedded in a Hilbert space. By isomorphisms of Hilbert spaces this is equivalent to an adversary that chooses an expert embedding matrix $U \in \mathbb{R}^{N \times N}$ such that for every u_i we have $\|u_i\|_2 \leq 1$ and correspondingly at each time point we receive a vector v_t such that $\|v_t\|_2 \leq 1$ as a result we have a factorization:

$$L = UV^T, \quad \|U\|_{2,\infty}, \|V\|_{2,\infty} \leq 1,$$

where $\|X\|_{2,\infty} = \sup_{\|y\| \leq 1} \|Xy\|_\infty$. Recall the definition of the max-norm, also called the γ_2 -norm (Srebro and Shraibman, 2005):

$$\|L\|_{\max} = \min_{UV^T=L} \|U\|_{2,\infty} \cdot \|V\|_{2,\infty}.$$

Thus, similar to the low rank setting we can define this setting as follows: At each round a learner chooses $x_t \in \Delta_N$, an adversary replies by choosing a loss vector ℓ_t , and the learner incurs the corresponding loss. The adversary is restricted to strategies such that $\|L\|_{\max} \leq 1$. The importance of this setting is that the proper generalization bound for this case is dimension independent (e.g., Kakade et al., 2009). Hence, we ask the following question:

Open Problem 2 *Is there an algorithm that can achieve regret $O(\sqrt{T})$ for any sequence ℓ_1, \dots, ℓ_T such that $\|L\|_{\max} \leq 1$?*

We can also generalize this setting to any pair of norms, $\|\cdot\|$ and its dual $\|\cdot\|^*$, where the description of the game remains the same. The adversary chooses an embedding U of the experts with bounded $\|\cdot\|$ norm. Then, at each round he chooses a set of vectors v_t with $\|\cdot\|^*$ bounded norm.

Finally, a different interesting direction to pursue in future work is to extend the noisy result to the adversarial setting. Namely,

Open Problem 3 *Is there an algorithm that can achieve regret $O(\sqrt{dT} + \epsilon\sqrt{T \log N})$ for any sequence ℓ_1, \dots, ℓ_T such that $\text{rank}_\epsilon(L) \leq d$?*

Acknowledgments

Elad Hazan acknowledges support from the National Science Foundation grant IIS-1523815. Roi Livni is a recipient of the Google Europe Fellowship in Learning Theory, and this research is supported in part by this Google Fellowship. Yishay Mansour is supported in part by the Israeli Centers of Research Excellence (I-CORE) program, (Center No. 4/11), by a grant from the Israel Science Foundation (ISF), by a grant from United States-Israel Binational Science Foundation (BSF) and by a grant from the Len Blavatnik and the Blavatnik Family Foundation. The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 336078 ERC-SUBLRN.

References

- Noga Alon, Troy Lee, Adi Shraibman, and Santosh Vempala. The approximate rank of a matrix and its algorithmic applications: approximate rank. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 675–684. ACM, 2013.
- Keith Ball. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58, 1997.
- Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*. Citeseer, 2009.
- Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.
- Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. 2006.
- Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.
- Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *COLT*, pages 6–1, 2012.
- Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. *arXiv preprint arXiv:1102.3923*, 2011.
- Eyal Gofer, Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for branching experts. In *Conference on Learning Theory*, pages 618–638, 2013.
- Andrew Goldberg, Ben Recht, Junming Xu, Robert Nowak, and Xiaojin Zhu. Transduction with matrix completion: Three birds with one stone. In *Advances in neural information processing systems*, pages 757–765, 2010.

- Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.
- Elad Hazan. *Introduction to Online Convex Optimization, Draft*. now Publishers Inc., 2015.
- Elad Hazan and Satyen Kale. On stochastic and worst-case models for investing. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 709–717. Curran Associates, Inc., 2009.
- Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.
- Elad Hazan and Satyen Kale. Better algorithms for benign bandits. *The Journal of Machine Learning Research*, 12:1287–1311, 2011.
- Elad Hazan, Roi Livni, and Yishay Mansour. Classification with low rank and missing data. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 257–266, 2015.
- Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.
- Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.
- Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013.
- Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In *Advances in Neural Information Processing Systems*, pages 1984–1992, 2010.
- Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Localization and adaptation in online learning. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 516–526, 2013.
- Sasha Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems*, pages 2141–2149, 2012.
- Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.
- Shai Shalev-Shwartz, Alon Gonen, and Ohad Shamir. Large-scale convex minimization with a low-rank constraint. *arXiv preprint arXiv:1106.1622*, 2011.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Learning Theory*, pages 545–560. Springer, 2005.

Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.

Appendix A. Technical Proofs

Lemma 11 *Let $M \in \mathbb{R}^{d \times d}$, $U \in \mathbb{R}^{d \times n}$ such that $M > 0$ and U . Then*

$$U(U^\top MU)^\dagger U^\top = M^{-1}.$$

Proof Let $N = M^{1/2}U$. Then, we have $N(N^\top N)^\dagger N^\top = I_d$. To see this, write the SVD decomposition $N = O\Sigma V^\top$ with diagonal non-singular $\Sigma \in \mathbb{R}^{d \times d}$ and $OO^\top = O^\top O = V^\top V = I_d$. Then,

$$N(N^\top N)^\dagger N^\top = O\Sigma V^\top (V\Sigma^2 V^\top)^\dagger V\Sigma O^\top = O\Sigma V^\top (V\Sigma^{-2} V^\top) V\Sigma O^\top = I_d.$$

Expanding the definition of N , we get $M^{1/2}U(U^\top MU)^\dagger U^\top M^{1/2} = I_d$, and since $M^{1/2}$ is non-singular, we can multiply by $M^{-1/2}$ on both sides and obtain the lemma. \blacksquare

A.1. Proof of Lemma 6

The proof relies on the following corollary of John’s Theorem:

Lemma 12 *Let K be a symmetric convex set centered around zero in \mathbb{R}^d . There exists a positive semi-definite matrix Σ such that for every $x \in K$:*

$$x^\top \Sigma x \leq \sup_{f \in K^*} |f(x)|^2 \leq d(x^\top \Sigma x).$$

Proof of Lemma 6 W.l.o.g., we assume $\alpha = 1$, the general case follows since $R(\alpha\mathcal{F}, S) = \alpha R(\mathcal{F}, S)$. We have

$$R(\mathcal{F}, S) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{t} \sum_{i=1}^t \sigma_i f(\ell_i) \right] = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} f\left(\frac{1}{t} \sum_{i=1}^t \sigma_i \ell_i\right) \right] \leq \sqrt{\mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} f^2\left(\frac{1}{t} \sum_{i=1}^t \sigma_i \ell_i\right) \right]}.$$

Next, we take Σ whose existence follows from Lemma 12. Note that Σ defines a scalar product. Specifically let us denote $\langle \ell_i, \ell_j \rangle = \ell_i^\top \Sigma \ell_j$, and also we let $\|\ell_i\|_2^2 = \ell_i^\top \Sigma \ell_i$. Then we have

$$\begin{aligned}
 \sqrt{\mathbb{E}_\sigma \left[\sup_{f \in K^*} f^2 \left(\frac{1}{t} \sum_{i=1}^t \sigma_i \ell_i \right) \right]} &\leq \sqrt{d \mathbb{E}_\sigma \left[\frac{1}{t^2} \left\| \sum_{i=1}^t \sigma_i \ell_i \right\|_2^2 \right]} \\
 &= \sqrt{d \mathbb{E}_\sigma \left[\frac{1}{t^2} \sum_{i,j=1}^t \sigma_i \sigma_j \langle \ell_i, \ell_j \rangle \right]} \\
 &= \sqrt{d \mathbb{E}_\sigma \left[\frac{1}{t^2} \sum_{i=1}^t \sigma_i^2 \|\ell_i\|_2^2 \right]} = \sqrt{\frac{d}{t^2} \sum_{i=1}^t \|\ell_i\|_2^2} \\
 &\leq \sqrt{\frac{d}{t} \max_i \|\ell_i\|_2^2} \leq \sqrt{\frac{d}{t} \max_i \sup_{f \in K^*} f^2(\ell_i)} \leq \sqrt{\frac{d}{t}},
 \end{aligned}$$

as claimed. ■

Appendix B. Lower Bounds for the AdaGrad Algorithm

AdaGrad (see Algorithm 3) is an algorithm that adapts the regularization matrix with respect to prior losses. Our aim in this section is to show that this learning scheme of the regularization cannot lead to a regret bound that is independent of the number of experts. Our strategy is as follows: since the AdaGrad algorithm depends on a learning rate parameter η we consider two cases: either η scales with N and becomes smaller, but then we show that for some sequence the algorithm's update is "too slow". On the other hand, we show that if η does not scale with N , the algorithm becomes less stable, and we can again inflict damage. Taken together we prove the following statement:

Theorem 13 *Consider Algorithm 3. For concreteness we assume that $x_1 = \frac{1}{N} \mathbf{1}$. For sufficiently large N , if $T < \sqrt{N}/6$ then there exist a sequence $\{\ell_1, \dots, \ell_T\}$ such that $\text{Regret}_T \geq T/2$ and $\text{rank}(L) = 1$.*

Lemma 14 *Consider Algorithm 3 with arbitrary η and δ . For concreteness we assume that $x_1 = \frac{1}{N} \mathbf{1}$. For sufficiently large N , if $T \leq \max\left(\frac{1}{36\eta^2} + 2\frac{\sqrt{\delta}}{6\eta}, \eta^2 N - \delta\right)$ then there exist a sequence ℓ_1, \dots, ℓ_T such that $\text{Regret}_T \geq T/2$ and $\text{rank}(L) = 1$.*

Proof We prove each bound separately.

Case 1: $T < \frac{1}{36\eta^2} + 2\frac{\sqrt{\delta}}{6\eta}$. We let $\ell_t = \mathbf{e} = (-1, \frac{1}{N-1}, \frac{1}{N-1}, \frac{1}{N-1}, \dots, \frac{1}{N-1})$ for all t . For every t we have that

$$G_t = \sqrt{t \mathbf{e} \mathbf{e}^\top + \delta I}$$

and

$$\eta G_t^{-1} \ell_t = \frac{\eta}{\sqrt{t + \delta \|\mathbf{e}\|}} \mathbf{e}.$$

Algorithm 3 AdaGrad

- 1: Input: $\eta, \delta, x_1 \in \Delta_N$.
- 2: Initialize: $S_0 = G_0 = \delta I$
- 3: **for** $t = 1$ to T **do**
- 4: Observe ℓ_t , suffer cost $x_t \cdot \ell_t$.
- 5: set

$$S_t = S_{t-1} + \ell_t \ell_t^\top, \quad G_t = S_t^{1/2}$$

$$y_{t+1} = x_t - \eta G_t^{-1} \ell_t$$

$$x_{t+1} = \arg \min_{x \in \Delta_N} \|y_{t+1} - x\|_{G_t}^2$$

6: **end for**

Next we use the inequality:

$$\sum_{t=1}^T \frac{1}{\sqrt{t+\delta}} \leq \int_0^T \frac{1}{\sqrt{t+\delta}} dt = 2 \left(\sqrt{T+\delta} - \sqrt{\delta} \right).$$

For $T \leq \left(\frac{1}{6\eta} + \sqrt{\delta} \right)^2 - \delta = \frac{1}{36\eta^2} + 2\frac{\sqrt{\delta}}{6\eta}$, we have that:

$$\sqrt{T+\delta} - \sqrt{\delta} < \frac{1}{6\eta};$$

$$\frac{1}{N} + \frac{\eta}{\|\mathbf{e}\|} \sum_{i=1}^T \frac{1}{\sqrt{t+\delta}} \leq \frac{1}{N} + \frac{2\eta}{\|\mathbf{e}\|} \left(\sqrt{T+\delta} - \sqrt{\delta} \right) \leq \frac{1}{2},$$

where last inequality follows since $\|\mathbf{e}\| > 1$ and we assume $N \geq 6$. One can observe that our update rule does not take y_t out of the simplex Δ_N and we have

$$x_t = \frac{1}{N} \mathbf{1} - \frac{\eta}{\|\mathbf{e}\|} \sum \frac{1}{\sqrt{t+\delta}} \mathbf{e},$$

and further, $x_t(1) < \frac{1}{2}$. In hindsight $x_t(1)$ suffers loss $-T$ while all other experts suffer positive loss. Hence the algorithm's regret is at least

$$\text{Regret}_T \geq \frac{T}{2}.$$

Case 2: $T < \eta^2 N - \delta$. We now choose $\mathbf{e} = \underbrace{(+1, +1, +1, +1, +1, +1)}_{N/2 \text{ times}}, \underbrace{(-1, -1, -1, -1, -1, -1)}_{N/2 \text{ times}}$.

and let

$$\ell_t = (-1)^{t+1} \mathbf{e}.$$

As before note that

$$\eta G_t^{-1} \ell_t = \frac{(-1)^{t+1} \eta}{\sqrt{t+\delta} \|\mathbf{e}\|} \mathbf{e} = \frac{(-1)^{t+1} \eta}{\sqrt{N(t+\delta)}} \mathbf{e}.$$

We claim that for $\sqrt{t+\delta} < \eta\sqrt{N}$ and $t > 1$ we have that:

$$x_t = \frac{2}{N} \begin{cases} \underbrace{(1, 1, 1, 1, 1, 1)}_{N/2 \text{ times}}, \underbrace{(0, 0, 0, 0, 0, 0)}_{N/2 \text{ times}} & t \text{ is even,} \\ \underbrace{(0, 0, 0, 0, 0, 0)}_{N/2 \text{ times}}, \underbrace{(1, 1, 1, 1, 1, 1)}_{N/2 \text{ times}} & t \text{ is odd.} \end{cases} \quad (6)$$

Hence $x_t \ell_t = 1$ and since the cumulative loss of each expert is at most 1 we have that:

$$\text{Regret}_T \geq \frac{T}{2}.$$

To see that Eq. (6) holds, we will show the statement for x_2 other cases are easier and follow the same proof: $y_1 = \mathbf{1} \frac{1}{N} + \alpha \mathbf{e}$, where $|\alpha| > \frac{1}{\sqrt{N}}$, hence it has the form

$$y_t = \underbrace{(a, a, a, a, a, a)}_{N/2 \text{ times}}, \underbrace{(-b, -b, -b, -b, -b, -b)}_{N/2 \text{ times}}$$

where $a - b = 2/N$ and $a, b > 0$. The statement now follows from Lemma 15 (see below). \blacksquare

Proof of Theorem 13 By Lemma 14 we need to show that

$$\min_{\eta, \delta} \max \left(\frac{1}{9\eta^2} + 2 \frac{\sqrt{\delta}}{3\eta}, \eta^2 N - \delta \right) > \frac{\sqrt{N}}{6}.$$

To prove this, we note that since both terms in the max are monotone in both variables, the minimum is attained when there is equality, i.e., the minimal η, δ satisfy:

$$\frac{1}{36\eta^2} + 2 \frac{\sqrt{\delta}}{6\eta} = \eta^2 N - \delta.$$

Since $\frac{1}{36\eta^2} + 2 \frac{\sqrt{\delta}}{6\eta} + \delta = \left(\frac{1}{6\eta} + \sqrt{\delta} \right)^2$, we get:

$$\sqrt{N} = \frac{1}{\eta} \left(\frac{1}{6\eta} + \sqrt{\delta} \right),$$

and we have that:

$$\frac{\sqrt{N}}{6} = \frac{\sqrt{2}}{36\eta^2} + \frac{\sqrt{\delta}}{6\eta} < \frac{\sqrt{2}}{36\eta^2} + 2 \frac{\sqrt{\delta}}{6\eta}.$$

It remains to prove Lemma 15, that was used for the proof of Lemma 14.

Lemma 15 Let $y = \underbrace{(a, a, a, a, a, a)}_{N/2 \text{ times}}, \underbrace{(-b, -b, -b, -b, -b, -b)}_{N/2 \text{ times}}$ where $a, b \geq 0$ and assume that $a - b = 2/N$. Let $G_t = \sqrt{\delta I + \alpha \mathbf{e} \mathbf{e}^\top}^{-1/2}$ for some $\alpha > 0$, where

$$\mathbf{e} = \underbrace{(+1, +1, +1, +1, +1, +1)}_{N/2 \text{ times}}, \underbrace{(-1, -1, -1, -1, -1, -1)}_{N/2 \text{ times}}.$$

Then

$$\min_{x \in \Delta_N} \frac{1}{2} \|y - x\|_{G_t}^2 = \frac{2}{N} \left(\underbrace{1, 1, 1, 1, 1, 1}_{N/2 \text{ times}}, \underbrace{0, 0, 0, 0, 0}_{N/2 \text{ times}} \right).$$

Proof Considering the Lagrangian and KKT conditions, we observe that x minimizes the distance iff the following hold:

1. $x \in \Delta_N$ (primal feasibility)
2. $\lambda > 0$ and $\theta(1) = \theta(2) = \dots = \theta(N)$. (dual feasibility)
3. $x = y + G_t^{-1}(\lambda + \theta)$ (stationarity)
4. $x(i) \neq 0 \Rightarrow \lambda(i) = 0$ and $\lambda(i) \neq 0 \Rightarrow x(i) = 0$. (complementary slackness)

Next note that e is an eigenvector of G_t and we have for some $c < 0$ that

$$G_t^{-1}ce = \left(\underbrace{-b, -b, -b, -b, -b, -b}_{N/2 \text{ times}}, \underbrace{+b, +b, +b, +b, +b}_{N/2 \text{ times}} \right).$$

Now we can write

$$ce = \underbrace{\left(\underbrace{0, 0, \dots, 0, 0}_{N/2 \text{ times}}, \underbrace{-2c, -2c, \dots, -2c, -2c}_{N/2 \text{ times}} \right)}_{\lambda} + \underbrace{\left(\underbrace{c, c, c, c, c, c}_{N/2 \text{ times}}, \underbrace{c, c, c, c, c, c}_{N/2 \text{ times}} \right)}_{\theta},$$

that concludes the proof. ■