New Algorithms for Repeated Play and Universal Portfolio Management

Amit Agarwal   Elad Hazan*

Abstract

We introduce a new algorithm, Smooth Prediction, and a new analysis technique that is applicable to a variety of online optimization scenarios, including regret minimization for Lipschitz regret functions studied by Hannan, *universal portfolios* by Cover, Kalai and Vempala, and others. Our algorithm is more efficient and applies to a more general setting (e.g. when the payoff function is unknown).

The algorithm extends a method proposed by Hannan in the 1950's, called 'Follow the Leader". It adds a barrier function to the convex optimization routine required to find the "leader" at every iteration.

One application of our algorithm is the well studied *universal portfolio management* problem, for which our algorithm is the first to combine optimal regret with computational efficiency. For more general settings, our algorithm achieves exponentially lower regret then previous algorithms.

1 Introduction

Consider the following general setting for repeated play. An online player chooses an action from a set of possible actions, without knowing the future. Nature then reveals a payoff for each possible action. This scenario is repeated T times. The player's goal is to minimize his total regret, which is defined to be the difference between his total payoff and the payoff of the best fixed action in hindsight.

In the basic model for online game playing, an online player $\mathcal{A}$ chooses a probability distribution p over a set of n possible actions (pure strategies). Nature then reveals a payoff $x(i) \in \mathbb{R}$ for each possible action. The expected payoff of the online player is $p^T x$ (we will abuse notation and denote this by px), where x is the n-dimensional payoff vector. This scenario is repeated for T iterations. If we denote the player's distribution at time $t \in [T]$ by p_t, and the payoff vector for time t by x_t, then the total payoff achieved by the online player is $\sum_{t=1}^{T} p_t x_t$. The payoff is compared to the maximum payoff attainable by a fixed distribution on pure strategies. This is captured by the notion of *regret* – the difference between the player's total payoff and the best payoff he could have achieved using a fixed distribution on pure strategies. Formally [1]:

$$R(\mathcal{A}) \triangleq \max_{p^* \in \mathcal{S}^n} \sum_{t=1}^{T} p^* x_t - \sum_{t=1}^{T} p_t x_t$$

The performance of an algorithm is measured by two parameters: the total regret and the time for the algorithm to compute the strategy p_T for iteration T.

*Supported by Sanjeev Arora's NSF grants MSPA-MCS 0528414, CCF 0514993, ITR 0205594

[1] $\mathcal{S}^n$ here denotes the n-dimensional simplex, i.e. the set of points $p \in \mathbb{R}^n$, $\sum_i p_i = 1, p_i \geq 0$

For the above basic setting, the regret is lower bounded by $\Omega(\sqrt{T})$ (for the models considered in this paper, T is the paramount complexity parameter and other parameters are usually treated as constants. The O notation hides factors independent of T). This regret has been achieved by several algorithms. The earliest were discovered in the game-theory and operations research community, and more efficient algorithms (Hedge and Adaboost by Freund and Schapire [FS97]) were discovered in machine learning. Since this model is very well understood, it will not be considered in this paper.

Instead, we consider a more general *online optimization* framework, related to that introduced by Zinkevich [Zin03]. In Zinkevich's framework, the online player chooses a point in some convex set, rather then just the simplex. The payoff functions allowed are arbitrary concave functions over the set. This model extends to even more general scenarios, in which every iteration a different concave function f_t is used, and even when these functions are unknown till Nature reveals them together with the payoff for the play iteration. The online player $\mathcal{A}$ wants to minimize the corresponding notion of regret, namely

$$R(\mathcal{A}) \triangleq \max_{p^* \in \mathcal{S}^n} \sum_{t=1}^{T} f_t(p^*) - \sum_{t=1}^{T} f_t(p_t) \tag{1}$$

An interesting problem covered by this framework is the problem of *universal portfolio management*, where the objective is to devise a dynamic portfolio the difference of whose returns to the best *fixed* portfolio in hindsight over T time periods is minimized.

Zinkevich proceeds to analyze a form of gradient ascent, and shows that the regret for this setting decreases as

$$R(\text{gradient ascent}) = O(G^2 \cdot \sqrt{T})$$

where G is an upper bound on the Euclidian norm of the gradient for the functions revealed throughout the game. He notes that in fact this general case can be derived by reducing to the basic setting (with linear payoffs) using first order Taylor approximation. Though obtaining regret of $O(\sqrt{T}$ is optimal for the linear case, Zinkevich's work leaves open the possibility that for the case of non-linear functions, with second derivative bounded away from zero, better convergence rates can be achieved.

In fact, this is not mere speculation since for the universal portfolio management problem, where $f_t(p_t) \equiv \log(p_t x_t)$ (for more details on how the logarithm function arises in the portfolio management context see Appendix C), Cover [Cov91] introduced the UNIVERSAL algorithm with regret bounded by $R(\text{UNIVERSAL}) = O(\log T)$.

This regret bound was shown to be asymptotically optimal in [OC98]. However, the running time of Cover's algorithm (i.e the time it takes to produce the distribution p_t given all prior payoffs) is exponential in the number of stocks - for n stocks and the T^{th} day the running time is $\Omega(T^n)$. Kalai and Vempala [KV03b] proposed a *randomized* implementation of Cover that runs in polynomial time, though the polynomial is quite large, namely $\Omega(n^7 T^8)$. This rather large running time, and the complexity of the techniques used therein, render the algorithm difficult to implement and benchmark, and to the best of our knowledge no such implementation exists to date.

1.1 Our Results

In this paper we consider a restricted version of Zinkevich's model, in which the point chosen by the online player is in the simplex, and the payoff functions $\{f_t\}$ have a bounded Hessian. Besides generalizing the basic online learning model, this formulation also generalizes the main motivation for this work - the well-studied universal portfolio management problem.

Our main contribution is a *deterministic* algorithm, called SMOOTH PREDICTION, that achieves the following regret bounds. If the functions $\{f_t\}$ throughout the repeated play are concave, have bounded gradient and Hessian, then

$$R(\text{SMOOTH PREDICTION}) = O(\log T)$$

For the universal portfolio management problem, under the standard "no-junk-bonds" assumption (see Appendix C), the regret bounds are $R(\text{SMOOTH PREDICTION}) \leq 4n\frac{1}{r^2}\log T$ [2]. The algorithm can be modified using the technique of Helmbold et al, such that even without the no-junk-bonds assumption, the regret is bounded by $O(T^{2/3})$.

SMOOTH PREDICTION has running time of $O(n^3 T)$, a significant improvement over previous methods. Preliminary experimental results indicate that SMOOTH PREDICTION performs as well as the MW algorithm by Helmbold et al. [HSSW96] which is based on the Multiplicative Weights Update Method (and is the current champion [HSSW96], also see subsection 1.3), and surpasses it in some cases. The final version will report on experimental results.

In order to analyze the performance of SMOOTH PREDICTION, we introduce a new potential function which takes into account the second derivative of the payoff functions. This is necessary, since the regret for linear payoff functions is bounded from below by $\Omega(\sqrt{T})$. This potential function is motivated by interior point algorithms, in particular the Newton method, and its analysis requires new algebraic techniques beyond the usual Multiplicative Weights Updates Method [AHK05] (an algorithmic technique which underlies most of the current algorithms). We believe these techniques may be useful for other learning and online optimization problems.

1.2 Connection to "Follow The Leader"

A natural algorithmic scheme in repeated play is called "Follow-The-Leader", first proposed by Hannan [Han57]. As the name suggests, the basic idea is to play the strategy that would have been optimal up to date. This does not ensure that the regret converges to zero in many simple scenarios, most obviously for linear payoff functions, but also for the universal portfolio management problem. However, as proposed and analyzed by Hannan, and recently analyzed by Kalai and Vempala [KV03a] (the original proof by Hannan is virtually unreadable), adding a small perturbation to the optimal strategy so far *does* ensure the regret converges to zero if the payoff functions are linear.

A natural question posed by Kalai and Vempala [KV03a] is whether a variant of Follow-The-Leader ensures small regret for the portfolio problem. From a computational viewpoint, this question is interesting since computing the best strategy in hindsight (and thus Follow-The-Leader), is known to be in solvable efficiently in polynomial time using interior point methods, see [NN94].

Merhav and Feder [MF92] show that Follow-The-Leader has optimal regret under some very strong assumptions. However, these assumptions do not even hold for the universal portfolio management problem, with or without the no-junk-bonds assumption [Fed].

Before each time period, SMOOTH PREDICTION computes the optimum of some convex program which is a "smoothened" version of the best mixed strategy in hindsight(i.e the "leader"). The smoothening is achieved by adding a logarithmic barrier function to the convex program for the best mixed strategy in hindsight. In particular, the barrier function prevents the optimum from lying too close to the boundary of the simplex.

Our analysis shows that this modification to Follow-The-Leader ensures low regret, thereby answering the question of Kalai and Vempala [KV03a].

[2]Here r is a lower bound on the price relative of any asset during one time period, i.e. $x_t(i) \geq r, \forall t, i$

1.3 Other related work

The MW algorithm by Helmbold et al. [HSSW96] mentioned earlier is very efficient to implement - the running time per time period is independent of T. Its regret is similar to the general bound of Zinkevich - $O(G^2 \cdot \sqrt{T})$. The gradient of the logarithmic payoff $\log(p_t x_t)$ is proportional to $\frac{1}{p_t x_t}$, and thus in order to upper bound G, one needs the "no-junk-bonds" assumption on the payoffs x_t (see Appendix C), i.e. $x_t(i) > r, \forall i \in [n]$. Under this assumption $R(MW) = O(\frac{1}{r^2}\sqrt{T})$.

Helmbold et al. [HSSW96] also present a modification to their algorithm which ensures that the regret is bounded by $O(T^{3/4})$ even without the "no-junk-bonds" assumption. The convergence analysis of their algorithm is tight as shown by [SL05].

2 Notation and Theorem Statements

The input is denoted by T vectors $(x_1, ..., x_T), x_t \in \mathbb{R}^n$ where $x_j(i)$ is the payoff of the i^{th} pure strategy during the j^{th} time period. We assume that $x_j(i) \leq 1, \forall i, j$. The x_t's have different interpretation depending on the specific application, but in general we refer to them as payoff vectors.

A (mixed) strategy is simply a fractional distribution over the pure strategies. We represent this distribution by $p \in \mathbb{R}^n$ where $\sum_i p_i = 1, p_i \geq 0$. So p is an element of the $(n-1)$-dimensional simplex. We assume that the payoff functions mapping distributions to real numbers, denoted by $f_t(px_t)$, are concave functions of the inner product, hence $f_t''(px_t) < 0$. Throughout the paper we assume the following about these functions:

1. $\forall t$, the payoffs are bounded by $0 \leq f_t(px_t) \leq \omega$ (positivity is w.l.o.g, as the shifting the payoff functions doesn't change the regret nor the following assumptions).

2. The $\{f_t\}$'s have bounded derivative $\forall t, p\ , |f_t'(px_t)| \leq G$.

3. The functions $\{f_t\}$ are concave with second derivative bounded from above by $f_t''(px_t) \leq -H < 0, \forall t$.

For a given set of T payoff vectors, $(x_1, ..., x_T), x_t \in \mathbb{R}^n_+$, we denote by $p^*(x_1, ..., x_T) = p^*$ the best distribution in hindsight, i.e.

$$p^* = \text{argmax}_p \{\sum_{t=1}^{T} f_t(px_t)\}$$

The Universal Portfolio Management problem can be phrased in the online game playing framework as described in Appendix C. The payoff at iteration t is $\log(p_t x_t)$, where x_t is the vector of price relatives.

Note that since $\log(c \cdot p_t x_t) = \log(c) + \log(p_t x_t)$, scaling the payoffs will only change the objective function by an additive constant making the objective invariant to scaling. Thus we can assume w.l.o.g that $\forall t$. $\max_{i \in [n]} x_t(i) = 1$ and $f_t(p_t x_t) \leq 1$. The "no-junk-bond" assumption implies $\forall t, i\ x_t(i) \geq r$, and thus $f_t'(p_t x_t) = \frac{1}{p_t x_t} \in [1, \frac{1}{r}]$, and similarly $f_t''(p_t x_t) = -\frac{1}{(p_t x_t)^2} \in [-\frac{1}{r^2}, -1]$.

2.1 Smooth Prediction

A formal definition of Smooth Prediction is as follows, where $e_i \in \mathbb{R}^n$ is the i'th standard basis vector (i.e. the vector that has zero in all coordinates but for the i'th, in which it is one)

SMOOTH PREDICTION

1. Let $\{f_1, ..., f_{t-1}\}$ be the concave payoff functions up to day t

   Solve the following convex program using interior point methods

$$\max_{p \in R^n} \left(\sum_{i=1}^{t-1} f_i(px_i) + \sum_{i \in n} \log(pe_i) \right) \tag{2}$$
$$\sum_{i=1}^{n} p_i = 1$$
$$\forall i \in [n] \ . \ p_i \geq 0$$

2. Play according to the computed distribution

We note the strategy of SMOOTH PREDICTION at time t by p_{t-1}. The performance guarantee for this algorithm is

Theorem 1 (main) *For any set of payoff vectors* $(x_1, ..., x_T)$

$$R(\text{SMOOTH PREDICTION}) \leq 4n \frac{G^2}{H} \log(\omega n T)$$

Corollary 2 *For the universal portfolio management problem, assuming the price relatives are lower bounded by* r*, for any set of price relative vectors* $(x_1, ..., x_T)$

$$R(\text{SMOOTH PREDICTION}) \leq 4n \frac{1}{r^2} \log(nT)$$

2.2 Running Time

Interior point methods [NN94] allow maximization of a n-dimensional concave function over a convex domain in time $\tilde{O}(n^{3.5})$. The most time consuming operations carried out by basic versions of these algorithms require computing the gradient and inverse Hessian of the function at various points of the domain. These operations require $O(n^3)$ time.

To generate the p_T at time T, SMOOTH PREDICTION maximizes a sum of $O(T)$ concave functions. Computing the gradient and inverse Hessian of such sum of functions can naturally be carried out in time $O(T \cdot n^3)$. All other operations are elementary and can be carried out in time independent of T. Hence, SMOOTH PREDICTION can be implemented to run in time $\tilde{O}(Tn^{3.5})$.

We note that in practice, many times approximations to p_T are sufficient. For benchmarking, we implement an efficient polynomial time approximation scheme using the ideas of Halperin and Hazan [HH05].

In section 4 we show that even without assuming the "no-junk-bonds" assumption, a modified version of SMOOTH PREDICTION has regret converging to zero. This modification predicts distributions which are a convex combination of SMOOTH PREDICTION's distribution and the uniform distribution (see section 4 for more detail).

Theorem 3 *For the universal portfolio management problem, for any set of price relative vectors* $(x_1, ..., x_T)$

$$R(\text{MODIFIED SMOOTH PREDICTION}) \leq 4n^3 T^{2/3}$$

We remark that similar results can be obtained for general concave regret functions in addition to the logarithmic function of the universal portfolio management problem. A general result of this nature for all concave functions satisfying certain conditions will be added in the full version of the paper.

3 Proof of Main Theorem

In this section we prove Theorem 1. The proof contains two parts: first, we compare Smooth Prediction to the algorithm Offline Prediction. The Offline Prediction algorithm is the same as Smooth Prediction except that it knows the payoff vector for the coming day in advance, i.e. on day t it plays according to p_t - the solution to convex program (2) with the payoff vectors $(x_1, \ldots, x_t)$. This part of the proof stated as Lemma 7 is similar in concept to the Kalai-Vempala result proved in subsection 3.1 henceforth.

The second part of the proof, constituting the main technical contribution of this paper, shows that Smooth Prediction is not much worse than Offline Prediction.

Lemma 4

$$\sum_{t=1}^{T} [f_t(p_t x_t) - f_t(p_{t-1} x_t)] \leq 4n \frac{G^2}{H} \cdot \log(nT)$$

Proof: Since p_t and p_{t-1} are the optimum distributions for period t and $t-1$, respectively, by Taylor expansion we have

$$\begin{aligned} f_t(p_t x_t) - f_t(p_{t-1} x_t) &= f_t'(p_{t-1} x_t)(p_t x_t - p_{t-1} x_t) + \frac{1}{2} f''(\zeta)(p_t x_t - p_{t-1} x_t)^2 \\ &\leq f_t'(p_{t-1} x_t)(p_t x_t - p_{t-1} x_t) = f_t'(p_{t-1} x_t) x_t^\top (p_t - p_{t-1}) \end{aligned} \quad (3)$$

for some ζ between $p_{t-1} x_t$ and $p_t x_t$. The inequality follows from the fact that f_t is concave and thus $f_t''(\zeta) < 0$. We proceed to bound the last expression by deriving an expression for $N_t \triangleq p_t - p_{t-1}$.

We claim that for any $t \geq 1$, p_t lies strictly inside the simplex. Otherwise, if for some $i \in [n]$ we have $p_t(i) = 0$, then $p_t e_i = 0$ and therefore the log-barrier term $f_0(p_t) = \sum_i \log(p_t e_i)$ approaches $-\infty$, whereas the return of the uniform distribution is positive which is a contradiction. We conclude that $\forall i \in [n] \ . \ p_t(i) > 0$ and therefore, p_t is strictly contained in the simplex. Hence according to convex program (2)

$$\nabla \log(p\mathcal{P}) \mid_{p=p_T} + \sum_{t=1}^{T} \nabla f_t(p x_t) \mid_{p=p_T} = \vec{0}$$

Applying the same considerations for p_{t-1} we obtain $\nabla \log(p\mathcal{P}) \mid_{p=p_{T-1}} + \sum_{t=1}^{T-1} \nabla f_t(p x_t)|_{p=p_{T-1}} = \vec{0}$. For notational convenience, denote $\log(p\mathcal{P}) = \sum_{i=1}^{n} \log(p e_i) \triangleq \sum_{t=-(n-1)}^{0} f_t(p x_t)$. Also note that $\nabla f_t(p x_t) = f_t'(p x_t) x_t$. From both observations we have

$$\sum_{t=-n+1}^{T} \left[f_t'(p_T x_t) x_t - f_t'(p_{T-1} x_t) x_t \right] = -f_T'(p_{T-1} x_T) x_T \quad (4)$$

By Taylor series, we have (for some ζ_T^t between $p_{t-1} x_t$ and $p_t x_t$)

$$\sum_{t=-n+1}^{T} f_t'(p_T x_t) = \sum_{t=-n+1}^{T} f_t'(p_{T-1} x_t) + \sum_{t=-n+1}^{T} \frac{1}{2} f_t''(\zeta_T^t)(p_T x_t - p_{T-1} x_t)$$

Plugging it back into equation (4) we get

$$\frac{1}{2}\sum_{t=-n+1}^{T} f_t''(\zeta_T^t) x_t x_t^\top N_t = \sum_{t=-n+1}^{T} [f_t'(p_T x_t) - f_t'(p_{T-1}x_t)] x_t = -f_T'(p_{T-1}x_T)x_T \tag{5}$$

This gives us a system of equations with the vector N_T as variables from which

$$N_T = 2\left(-\sum_{t=-n+1}^{T} f_t''(\zeta_T^t) x_t x_t^\top\right)^{-1} \cdot x_T f_T'(p_{T-1}x_T) \tag{6}$$

Let $A_t = -\sum_{i=-n+1}^{t} f_i''(\zeta_t^i) x_t x_t^\top$.

Now the regret can be bounded by (using equation (3)):

$$\begin{aligned}\sum_{t=1}^{T}[f_t(p_t x_t) - f_t(p_{t-1}x_t)] &\leq \sum_{t=1}^{T} f_t'(p_{t-1}x_t) x_t^\top N_t \\ &\quad \text{by previous bound on } N_t \\ &= 2\sum_{t=1}^{T}(f_t'(p_{t-1}x_t))^2 \cdot x_t^\top \left(-\sum_{i=-n+1}^{t} f_i''(\zeta_t^i) x_t x_t^\top\right)^{-1} x_t \\ &\leq 2G^2 \sum_{t=1}^{T} x_t^\top A_t^{-1} x_t\end{aligned}$$

The following lemma is proved in Appendix B.

Lemma 5 *For any set of rank 1 PSD matrices $Y_1, ..., Y_t$ and constants $\beta_1, ..., \beta_t \geq 1$ we have:*

$$(\sum_{i=1}^{t} \beta_i Y_i)^{-1} \leq (\sum_{i=1}^{t} Y_i)^{-1}$$

Let the matrix $C_T = \sum_{t=-n+1}^{T-1} x_t x_t^\top$. Applying Lemma 5 with $\beta_i = -f_i''(\zeta_t^i) \cdot \frac{1}{H}$ and $Y_i = C_t \cdot H$ implies that $\forall t \, . \, A_t^{-1} \leq \frac{1}{H} C_t^{-1}$.

Now back to bounding the regret, we have:

$$\sum_{t=1}^{T}[f_t(p_t x_t) - f_t(p_{t-1}x_t)] \leq \frac{2G^2}{H}\sum_{t=1}^{T} x_t^\top C_t^{-1} x_t = \frac{2G^2}{H}\sum_{t=1}^{T} C_t^{-1} \bullet x_t x_t^\top$$

To continue, we use the following lemma:

Lemma 6 *For any set of rank 1 PSD matrices $Y_1, ..., Y_T \in R^{n\times n}$ such that $\sum_{i=1}^{k-1} Y_i$ is invertible, we have*

$$\sum_{t=k}^{T}(\sum_{i=1}^{t} Y_i)^{-1} \bullet Y_t \leq \log \frac{|\sum_{t=1}^{T} Y_t|}{|\sum_{t=1}^{k-1} Y_t|}$$

Since $C_t = \sum_{i=-n+1}^{t-1} x_i x_i^T$, by the Lemma above

$$\sum_{t=n+1}^{T} [f_t(p_t x_t) - f_t(p_{t-1} x_t)] \quad \leq \quad 2\frac{G^2}{H} \log \frac{|\sum_{t=-n+1}^{T} x_t x_t^\top|}{|\sum_{t=-n+1}^{0} x_t x_t^\top|}$$

Recall that by the definition of Smooth Prediction and $\{f_t | t \in [-n+1, 0]\}$, we have that $\sum_{t=-n+1}^{0} x_t x_t^\top = I_n$, where I_n is the n-dimensional identity matrix. In addition, since every entry $x_t(i)$ is bounded in absolute value by 1, we have that $|(\sum_{t=-n+1}^{T} x_t x_t^\top)(i,j)| \leq T+1$, and therefore $|\sum_{t=-n+1}^{T} x_t x_t^\top| \leq n!(T+1)^n$. Plugging that into the previous expression we obtain

$$\sum_{t=1}^{T} [f_t(p_t x_t) - f_t(p_{t-1} x_t)] \quad \leq \quad 4\frac{G^2}{H} \log(n! T^n) \leq 4\frac{G^2}{H}(n \log T + n \log n)$$

This completes the proof of Lemma 4. □

Theorem 1 now follows from Lemma 7 and Lemma 4.

3.1 Proof of Lemma 7

Lemma 7

$$\sum_{t=1}^{T} [f_t(p^* x_t) - f_t(p_t x_t)] \leq 2n \log(nT\omega)$$

In what follows we denote $f_t(p) = f_t(p x_t)$, and let $f_0(p) = \sum_{i=1}^{n} \log(p e_i)$ denote the log-barrier function. Lemma 7 follows from the following two claims.

Claim 1

$$\sum_{t=0}^{T} f_t(p_t) \geq \sum_{t=0}^{T} f_t(p_T)$$

Proof: By induction on t. For $t = 1$ this is obvious, we have equality. The induction step is as follows:

$$\begin{aligned}
\sum_{t=0}^{T} f_t(p_t) &= \sum_{t=0}^{T-1} f_t(p_t) + f_T(p_T) \\
&\quad \text{by the induction hypothesis} \\
&\geq \sum_{t=0}^{T-1} f_t(p_{T-1}) + f_T(p_T) \\
&\quad \text{by definition of } p_T \\
&\geq \sum_{t=1}^{T-1} f_t(p_T) + f_T(p_T) \\
&= \sum_{t=0}^{T} f_t(p_T)
\end{aligned}$$

□

Claim 2

$$\sum_{t=1}^{T}[f_t(p^*) - f_t(p_T)] \leq 2n\log(T\omega) + f_0(p^*)$$

Proof: By the definition of p_T, we have:

$$\forall \hat{p} \ . \ \sum_{t=0}^{T} f_t(p_T) \ \geq \ \sum_{t=0}^{T} f_t(\hat{p})$$

In particular, take $\hat{p} = (1-\alpha)p^* + \frac{\alpha}{n}\vec{1}$ and we have

$$\begin{aligned}
\sum_{t=0}^{T} f_t(p_T) - \sum_{t=0}^{T} f_t(p^*) \ &\geq \ \sum_{t=0}^{T} f_t((1-\alpha)p_T^* + \frac{\alpha}{n}\vec{1}) - \sum_{t=0}^{T} f_t(p^*) \\
&\quad \text{since } f_t \text{ are concave and } f_0 \text{ is monotone} \\
&\geq \ (1-\alpha)\sum_{t=1}^{T} f_t(p^*) + \frac{\alpha}{n}\sum_{t=1}^{T} f_t(\vec{1}) + f_0(\frac{\alpha}{n}) - \sum_{t=0}^{T} f_t(p^*) \\
&\quad \text{the functions } f_t \text{ are positive} \\
&\geq \ -\alpha T\omega + n\log\frac{\alpha}{n} - f_0(p^*) \geq -2n\log(T\omega) - f_0(p^*)
\end{aligned}$$

Where the last inequality follows by taking $\alpha = \frac{n\log(T\omega)}{T\omega}$. $\square$

Lemma 7 now follows as a corollary:

Proof:[Lemma 7] Combining the previous two claims:

$$\begin{aligned}
\sum_{t=1}^{T}[f_t(p^*) - f_t(p_t)] \ &= \ \sum_{t=0}^{T}[f_t(p^*) - f_t(p_t)] - f_0(p^*) + f_0(p_0) \\
&\leq \ \sum_{t=0}^{T}[f_t(p^*) - f_t(p_T)] - f_0(p^*) + f_0(p_0) \\
&\leq \ 2n\log T + f_0(p_0)
\end{aligned}$$

To complete the proof, note that $p_0 = \frac{1}{n}\vec{1}$, and hence $f_0(p_0) = n\log\frac{1}{n}$. $\square$

4 Application to Universal Portfolio Management

In many cases, if an algorithm is universal (has regret converging to zero) under the no-junk-bonds assumption, then it is universal without this assumption . The reduction, due to Helmbold et al [HSSW96], consists of adding a small multiple of the uniform portfolio to the portfolios generated by the algorithm at hand. This has the effect that the return of the portfolio chosen is bounded from below, which suffices for proving universality for many algorithms.

In this section we prove that the same modification to Smooth Prediction is universal in the general case, using similar techniques as [HSSW96].

Proof:[Theorem 3]

For some $\alpha > 0$ to be fixed later, define the portfolio $\bar{p}_t \triangleq (1-\alpha)p_t + \alpha \cdot \frac{1}{n}\vec{1}$, i.e the portfolio which is a convex combination of Smooth Prediction's strategy at time t and the uniform portfolio. Similarly, let $\bar{x}_t = (1-\frac{\alpha}{n})x_t + \frac{\alpha}{n}\vec{1}$ be a "smoothened" price relative vector.

Then we have

$$\begin{aligned}
\log(\frac{\bar{p}_t x_t}{p_t \bar{x}_t}) &= \log \tfrac{(1-\alpha)p_t x_t + \frac{\alpha}{n} x_t \cdot \vec{1}}{(1-\frac{\alpha}{n})p_t x_t + \frac{\alpha}{n} p_t \cdot \vec{1}} \\
&\geq \log \tfrac{(1-\alpha)p_t x_t + \frac{\alpha}{n}}{(1-\frac{\alpha}{n})p_t x_t + \frac{\alpha}{n}} && \text{since } \max_j x_t(j) = 1 \\
&\geq \log((1-\alpha) + \tfrac{\alpha}{n}) \\
&\geq -2\alpha && \text{for } \alpha \in (0, \tfrac{1}{2})
\end{aligned}$$

Note that for every p and x_t we have $p\bar{x}_t = (1 - \frac{\alpha}{n})px_t + \frac{\alpha}{n} \geq \frac{\alpha}{n}$. Hence, by Corollary 2

$$\frac{1}{T}\sum_{t=1}^{T} \log(\frac{p^* \bar{x}_t}{p_t \bar{x}_t}) \leq \frac{6n}{(\alpha/n)^2} \frac{\log(nT)}{T}$$

Note that for every p, and in particular for p^*, it holds that $p\bar{x}_t = (1 - \frac{\alpha}{n})px_t + \frac{\alpha}{n} \geq px_t$. Combining all previous observations

$$\sum_{t=1}^{T} \log \frac{p^* x_t}{\bar{p}_t x_t} \leq \sum_{t=1}^{T} \log \frac{p^* \bar{x}_t}{\bar{p}_t x_t} = \sum_{t=1}^{T} \log(\frac{p^* \bar{x}_t}{p_t \bar{x}_t} \cdot \frac{p_t \bar{x}_t}{\bar{p}_t x_t}) \leq 6n^3 \alpha^{-2} \log(nT) + 2T\alpha$$

Choosing $\alpha = T^{-1/3}$ yields the result. □

5 Conclusions

In subsequent work with Adam Kalai and Satyen Kale we extend the techniques hereby and obtain variants of follow-the-leader which attain logarithmic regret in the general Zinkevich *online optimization* framework. The efficiency of these algorithms can further improved to run in time $O(n^2)$ per iteration. Other extensions include a variant of SMOOTH PREDICTION with logarithmic *internal regret* (a stronger notion of regret, see [SL05]).

6 Acknowledgements

We thank Sanjeev Arora and Rob Schapire for their insightful comments and advice. We thank Satyen Kale for proofreading an earlier version of this manuscript and providing a much simplified proof of Lemma 6, which is included hereby with his permission. The second author would like to thank Meir Feder for helpful correspondence.

References

[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.

[Bro05] M. Brookes. The matrix reference manual. *[online] http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html*, 2005.

[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.

[Cov96] T. M. Cover. Universal data compression and portfolio selection. In *FOCS '96: Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, page 534, Washington, DC, USA, 1996. IEEE Computer Society.

[Fed] Meir Feder. personal communications, june 2005.

[FKM05] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[FS97] Yoav Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[Han57] James Hannan. Approximation to bayes risk in repeated play. *In M. Dresher, A. W. Tucker, and P. Wolfe, editors, Contributions to the Theory of Games, volume III*, pages 97–139, 1957.

[HH05] Eran Halperin and Elad Hazan. Haplofreq: efficient algorithms for online decision problems. In *RECOMB*, pages 1–1, 2005.

[HSSW96] D. Helmbold, R. Schapire, Y. Singer, and M. Warmuth. On-line portfolio selection using multiplicative updates. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

[KV03a] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. In *16th Annual Conference on Computational Learning Theory*, pages 26–40, 2003.

[KV03b] Adam Kalai and Santosh Vempala. Efficient algorithms for universal portfolios. *J. Mach. Learn. Res.*, 3:423–440, 2003.

[MF92] Neri Merhav and Meir Feder. Universal sequential learning and decision from individual data sequences. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 413–427, New York, NY, USA, 1992. ACM Press.

[NN94] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications.* Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[OC96] Erik Ordentlich and Thomas M. Cover. On-line portfolio selection. In *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*, pages 310–313, New York, NY, USA, 1996. ACM Press.

[OC98] Erik Ordentlich and Thomas M. Cover. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research*, 23(4):960–982, 1998.

[SL05] G. Stoltz and G. Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59:125–159, 2005.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

A Proof of Lemma 6

First we require the following claim.

Claim 3 *For any PSD matrices A, B we have*

$$B^{-1} \bullet A \leq \log \frac{|B|}{|B-A|}$$

PROOF:

$$\begin{aligned}
B^{-1} \bullet A &= \mathbf{Tr}(B^{-1}A) && \because A \bullet B = \mathbf{Tr}(AB) \\
&= \mathbf{Tr}(B^{-1}(B-(B-A))) \\
&= \mathbf{Tr}(B^{-1/2}(B-(B-A))B^{-1/2}) && \because \mathbf{Tr}(AB) = \mathbf{Tr}(A^{-1/2}BA^{-1/2}) \\
&= \mathbf{Tr}(I - B^{-1/2}(B-A)B^{-1/2}) \\
&= \textstyle\sum_{i=1}^n \left[1 - \lambda_i(B^{-1/2}(B-A)B^{-1/2})\right] && \because \mathbf{Tr}(A) = \sum_{i=1}^n \lambda_i(A) \\
&\leq \textstyle\sum_{i=1}^n \log\left[\lambda_i(B^{-1/2}(B-A)B^{-1/2})\right] && \because 1 - x \leq -\log(x) \\
&= -\log\left[\textstyle\prod_{i=1}^n \lambda_i(B^{-1/2}(B-A)B^{-1/2})\right] \\
&= -\log|B^{-1/2}(B-A)B^{-1/2}| = \log \tfrac{|B|}{|B-A|} && \because \prod_{i=1}^n \lambda_i(A) = |A|
\end{aligned}$$

□

Lemma 6 now follows as a corollary:
PROOF:[Lemma 6] By the previous claim, we have

$$\begin{aligned}
\sum_{t=k}^{T} (\sum_{i=1}^{t} Y_i)^{-1} \bullet Y_t &\leq \sum_{t=k}^{T} \log \frac{|\sum_{i=1}^t Y_i|}{|\sum_{t=1}^t Y_i - Y_t|} \\
&= \log \frac{|\sum_{t=1}^T Y_t|}{|\sum_{t=1}^{k-1} Y_t|}
\end{aligned}$$

□

B Proof of Lemma 5

We denote for matrices $A \geq B$ if and only if $A - B \succeq 0$. AB denotes the usual matrix product, and $A \bullet B = trace(AB)$.

Claim 4 *For any constant $c \geq 1$ and psd matrices $A, B \geq 0$, such that B is rank 1, it holds that*

$$(A + cB)^{-1} \leq (A + B)^{-1}$$

Proof: By the Matrix Inversion Lemma [Bro05], we have that

$$(A+B)^{-1} = A^{-1} - \frac{A^{-1}BA^{-1}}{1 + A^{-1} \bullet B}$$

$$(A+cB)^{-1} = A^{-1} - \frac{cA^{-1}BA^{-1}}{1 + cA^{-1} \bullet B}$$

Hence, it suffices to prove:

$$\frac{cA^{-1}BA^{-1}}{1 + cA^{-1} \bullet B} \geq \frac{A^{-1}BA^{-1}}{1 + A^{-1} \bullet B}$$

Which is equivalent to (since A is psd, and all numbers are positive):

$$(1 + A^{-1} \bullet B)(cA^{-1}BA^{-1}) \geq (1 + cA^{-1} \bullet B)(A^{-1}BA^{-1})$$

And this reduces to:

$$(c-1)A^{-1}BA^{-1} \geq 0$$

which is of course true. □

Lemma 5 follows as a corollary of this claim.

C Universal Portfolio Management

A constant rebalanced portfolio (CRP) is an investment strategy which keeps the same distribution of wealth among a set of stocks from period to period. That is, the proportion of total wealth in a given stock is the same at the beginning of each period. Recently there has been work on on-line investment strategies which are competitive with the best CRP determined in hindsight [OC96, Cov91, CO96, Cov96, KV03b, BK97, HSSW96], in the sense that the daily performance of these algorithms approaches that of the best CRP as the number of periods grows without bound.

As an example of a useful CRP, consider the following market with just two stocks. The price of both stocks alternately halves and doubles, but with inverse correlation - when one stock halves the other doubles. Investing in a single stock will not increase the wealth by more than a factor of two. However, a $(\frac{1}{2}, \frac{1}{2})$ CRP will increase its wealth exponentially. At the end of each day it trades stock so that it has an equal worth in each stock. Each day, the total worth is increased by a factor of $\frac{5}{4}$.

Consider a market with n stocks and T days. The obvious parameter to measure the performance of a given investment algorithm is the (normalized) rate at which it accumulates wealth compared to the best CRP in hindsight, formally

$$\frac{1}{T} \log \left(\frac{\text{wealth of } \mathcal{A}}{\text{wealth of best CRP}} \right)$$

This performance measure fits naturally to the repeated play model. The wealth of the online player after T trading days is the initial wealth multiplied by $\prod_{t=1}^{T} p_t x_t$, where p_t is the portfolio at time t and x_t is the price relatives vector, i.e. $x_t(i)$ is the ratio of the prices of stock i at trading days t and $t-1$ (see [Cov91] for more details on this model). Hence, the logarithm of the wealth fits to the repeated play model where the concave function applied each iteration is simply $f_t(p) = \log(px_t)$. With these payoff functions, the performance measure above is simply the *regret* (normalized by the number of iterations).

An investment algorithm $\mathcal{A}$ is called *universal* if $\frac{1}{T}R(\mathcal{A}) \mapsto 0$ as $T \mapsto \infty$. The **"no-junk-bonds" assumption** asserts that the price relatives are bounded away from zero.