Electronic Colloquium on Computational Complexity, Report No. 33 (2006)

Efficient Algorithms for Online Game Playing and Universal Portfolio Management

Amit Agarwal Elad Hazan*

Abstract

A natural algorithmic scheme in online game playing is called 'follow-the-leader', first proposed by Hannan in the 1950's. Simply stated, this method advocates the use of past history to make future predictions, by using the optimal strategy so far as the strategy for the next game iteration. Randomized variations on this method for the special case of linear payoff functions have been rigorously analyzed and have found numerous applications in machine learning and game theory. It was a long standing open problem whether the 'follow the leader' method attains any non-trivial regret guarantees for the case of concave regret functions. This question is significant since 'follow-the-leader' is a natural *deterministic* method, easy to implement and computationally efficient.

We introduce a new analysis technique and show that a deterministic variant of this method has optimal regret. This result is applicable to a variety of online optimization scenarios, including regret minimization for Lipschitz regret functions, universal portfolios management, online convex optimization and online utility maximization. For the well studied *universal portfolio management* problem, our algorithm combines optimal regret with computational efficiency. For the general setting, our algorithm achieves exponentially lower regret than previous algorithms.

Our analysis shows a surprising connection between interior point methods and online optimization using follow-the-leader.

1 Introduction

We consider the following basic model for online game playing: the player $\mathcal{A}$ chooses a probability distribution p over a set of n possible actions (pure strategies) without knowing the future. Nature then reveals a payoff $x(i) \in \mathbb{R}$ for each possible action. The expected payoff of the online player is $f(p^\top x)$ (we will abuse notation and denote this by $f(px)$), where x is the n-dimensional payoff vector and f is a concave payoff function. This scenario is repeated for T iterations. If we denote the player's distribution at time $t \in [T]$ by p_t, the payoff vector by x_t and the payoff function by f_t, then the total payoff achieved by the online player is $\sum_{t=1}^{T} f_t(p_t x_t)$. 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]:

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

The performance of an online game playing 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 $p \in \mathbb{R}^n, \sum_i p_i = 1, p_i \geq 0$

ISSN 1433-8092

When the f_t's are linear functions, the regret is lower bounded by $\Omega(\sqrt{T})$. 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.

A well-studied problem covered by this framework is the problem of *universal portfolio management*, where the concave payoff function applied each iteration is $f_t(px) = \log(px)$ (see subsection 1.2). For this special case online game playing algorithms that break the $\Omega(\sqrt{T})$ regret lower bound have been devised by Cover [Cov91], who obtains an algorithm with only $O(\log T)$ regret. The caveat of Cover's algorithm is its running time which is exponential in the dimension, namely $\Omega(T^n)$. The running time was subsequently greatly improved by Kalai and Vempala [KV03] to $O(n^7T^8)$ albeit introducing randomization.

In this paper we propose and analyze an algorithm that combine the benefits of previous approaches: both computational efficiency and logarithmic regret. The algorithm applies for the more general case of online game playing, even when every iteration a different concave payoff function f_t is used, and when these functions are unknown till Nature reveals them together with the payoff for the play iteration.

1.1 Follow the leader

As the name suggests, the basic idea behind the "follow-the-leader" method is to play the strategy that would have been optimal up to date. The method was initially proposed and analyzed by Hannan [Han57], and recently revisited by Kalai and Vempala [KV05], who show that adding a small perturbation to the optimal strategy so far ensures $O(\sqrt{T})$ regret.

A natural question, asked by Larson [Lar86], Ordentlich [Ord96] and most recently Kalai and Vempala [KV05], is: *What is the performance of FTL for concave regret functions?* Hannan [Han57], and later 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, even when the price relatives are lower bounded [Fed].

In this paper we analyze a natural variant of follow-the-leader, which we call Smooth Prediction. 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. We show that this follow-the-leader variant ensures logarithmic regret for strictly concave payoff functions, thereby answering the question above. Smooth Prediction is a *deterministic* algorithm that has running time of $O(n^3T)$, a significant improvement over previous methods.

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] (the algorithmic technique underlying many online algorithms). We believe these techniques may be useful for other learning and online optimization problems.

1.2 Universal portfolio management

A well-studied problem which is covered by the framework considered in the paper is the problem of *universal portfolio management*, where the objective is to devise a dynamic portfolio the difference of whose returns to the best constant rebalanced portfolio (CRP)[2] in hindsight over T time periods is minimized. For a market with n stocks and T days, the regret function becomes

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

(see [Cov91] for more details). Hence, the logarithm of the wealth fits to the online game playing model where the concave function applied each iteration is simply $f_t(px) = \log(px)$.

On-line investment strategies competitive with the best CRP determined in hindsight have been devised using many different techniques. Cover et al. [OC96, Cov91, CO96, Cov96] proposed an exponential weighting scheme that attains optimal logarithmic regret. 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 [KV03] used general techniques for sampling logconcave functions over the simplex to devise a *randomized* polynomial time algorithm with a running time of $\Omega(n^7 T^8)$.

Helmbold et al. [HSSW96] used the general multiplicative updates method to propose an extremely fast portfolio management algorithm. However, the regret attained by their algorithm is bounded by $O(\sqrt{T})$ (as opposed to logarithmic regret), and that is assuming the "bounded variability" assumption (which states that the changes in commodity value are bounded, see section 2). The performance analysis of their algorithm is tight as shown by [SL05].

The Smooth Prediction algorithm analyzed hereby applies to the special case of universal portfolio management. Under the "bounded variability" assumption, it attains optimal $O(\log T)$ regret. The algorithm can be modified using the technique of Helmbold et al, such that the regret remains sublinear even without the bounded variability assumption.

1.3 Other related work

The online game playing framework we consider is somewhat more restricted than the *online convex optimization* framework of 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. We suspect that the techniques developed hereby can be applied to the full Zinkevich model [3].

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:

[2] A constant rebalanced portfolio 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.

[3] this indeed was achieved in subsequent work [HKKA06]

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 follows (see [KV03] for more details). The payoff at iteration t is $\log(p_t x_t)$, where p_t is the distribution of wealth on trading day t, and x_t is the vector of price relatives, i.e. the i'th entry is the ratio between the price of commodity i in day t and $t-1$.

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 "bounded variability" assumption states $\forall t, i$ $x_t(i) \geq r$, which translates to a bound on the price relatives - i.e the change in price for every commodity and trading day is bounded. This implies that the derivative of the payoff functions is bounded by $f'_t(p_t x_t) = \frac{1}{p_t x_t} \in [1, \frac{1}{r}]$, and similarly $f''_t(p_t x_t) = -\frac{1}{(p_t x_t)^2} \in [-\frac{1}{r^2}, -1]$.

We denote for matrices $A \geq B$ if and only if $A - B \succeq 0$, i.e the matrix $A - B$ is positive semi-definite (has only non-negative eigenvalues). AB denotes the usual matrix product, and $A \bullet B = \mathbf{Tr}(AB)$.

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

$$\max_{p \in R^n} \left(\sum_{i=1}^{t-1} f_i(px_i) + \sum_{i \in n} \log(pe_i) \right) \tag{1}$$

$$\sum_{i=1}^{n} p_i = 1$$

$$\forall i \in [n] \ . \ p_i \geq 0$$

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* $(x_1, ..., x_T)$

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

In section 4 we show that even without assuming the "bounded variability" assumption, a modified version of Smooth Prediction has sublinear regret. This modification predicts distributions which are a convex combination of Smooth Prediction's distribution and the uniform distribution (see section 4 for more detail).

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, such as the efficient polynomial time approximation scheme of Halperin and Hazan [HH05].

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 (1) with the payoff vectors $(x_1, \ldots, x_t)$. This part of the proof stated as Lemma 6 is similar in concept to the Kalai-Vempala result, and 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 3

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

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

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

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 (1)

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

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

$$\sum_{t=-n+1}^{T} \left[f_t'(p_T x_t)x_t - f_t'(p_{T-1}x_t)x_t \right] = -f_T'(p_{T-1}x_T)x_T \tag{3}$$

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 (3) we get

$$\frac{1}{2} \sum_{t=-n+1}^{T} f_t''(\zeta_T^t)x_t x_t^\top N_t = \sum_{t=-n+1}^{T} [f_t'(p_T x_t) - f_t'(p_{T-1}x_t)]x_t = -f_T'(p_{T-1}x_T)x_T \tag{4}$$

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

$$N_T = 2 \left(- \sum_{t=-n+1}^{T} f_t''(\zeta_T^t)x_t x_t^\top \right)^{-1} \cdot x_T f_T'(p_{T-1}x_T) \tag{5}$$

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 (2)):

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

The following lemma is proved in Appendix B.

Lemma 4 *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 4 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:

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

To continue, we use the following lemma, which is proved in Appendix A

Lemma 5 *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 3. □

Theorem 1 now follows from Lemma 6 and Lemma 3.

3.1 Proof of Lemma 6

Lemma 6

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

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 6 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 $\alpha = \frac{n \log(T\omega)}{T\omega}$. □

Lemma 6 now follows as a corollary:

PROOF:[Lemma 6] 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 $f_0(p_0) = n\log\frac{1}{n}$. □

4 Application to Universal Portfolio Management

In many cases, if an algorithm is universal (has sublinear regret) under the "bounded variability" 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].

Theorem 7 *For the universal portfolio management problem, for any set of price relative vectors* $(x_1, ..., x_T)$

$$R(\text{Modified Smooth Prediction}) \leq 5nT^{2/3}$$

Proof:

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

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

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

$$\sum_{t=1}^{T} \log(\frac{p^* \bar{x}_t}{p_t \bar{x}_t}) \leq \frac{4n}{(\alpha/n)^2} \log(nT)$$

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

$$\sum_{t=1}^{T} \log \frac{p^* x_t}{\bar{p}_t x_t} \leq \sum_{t=1}^{T} \log \frac{p^* \bar{x}_t}{\bar{p}_t x_t} = \sum_{t=1}^{T} \log(\frac{p^* \bar{x}_t}{p_t \bar{x}_t} \cdot \frac{p_t \bar{x}_t}{\bar{p}_t x_t}) \leq 4n^3 \alpha^{-2} \log(nT) + 2T\alpha$$

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

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 will be added in the full version of the paper.

5 Conclusions

In subsequent work with Adam Kalai and Satyen Kale [HKKA06] we extend the techniques in this paper to obtain variants of follow-the-leader which attain logarithmic regret in the general Zinkevich *online optimization* framework. The running time can be further improved to $O(n^2)$ per iteration. Other extensions include a variant that attains 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 5, 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.

[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.

[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.

[HKKA06] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. *manuscript*, 2006.

[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.

[KV03] Adam Kalai and Santosh Vempala. Efficient algorithms for universal portfolios. *J. Mach. Learn. Res.*, 3:423–440, 2003.

[KV05] Adam Kalai and Santosh Vempala. Efficient algorithms for on-line optimization. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[Lar86] David C. Larson. Growth optimal trading strategies. *Ph.D. dissertation, Stanford Univ., Stanford, CA*, 1986.

[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.

[Ord96] Erik Ordentlich. Universal investment and universal data compression. *Ph.D. dissertation, Stanford Univ., Stanford, CA*, 1996.

[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 *Proceedings of the Twentieth International Conference (ICML)*, pages 928–936, 2003.

A Proof of Lemma 5

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:

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

□

Lemma 5 now follows as a corollary:

PROOF:[Lemma 5] By the previous claim, we have

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

□

B Proof of Lemma 4

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

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

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:

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

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

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

And this reduces to:

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

which is of course true. □

Lemma 4 follows as a corollary of this claim.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/