Efficient Algorithms for Online Game Playing and Universal Portfolio Management

Amit Agarwal* Elad Hazan†

Abstract

We introduce a new algorithm and a new analysis technique that is applicable to a variety of online optimization scenarios, including regret minimization for Lipschitz regret functions, universal portfolio management, online convex optimization and online utility maximization. In addition to being more efficient and deterministic, our algorithm applies to a more general setting (e.g. when the payoff function is unknown). For the general online game playing setting it is the first to attain logarithmic regret, as opposed to previous algorithms attaining polynomial regret.

The algorithm extends a natural online method studied in the 1950's, called "follow the leader", thus answering in the affirmative a conjecture about universal portfolios made by Cover and Ordentlich and independently by Kalai and Vempala. The techniques also leads to derandomization of an algorithm by Hannan, and Kalai and Vempala.

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

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]:

$$\text{Regret}(\mathcal{A}) \triangleq \sup_{f_1,\ldots,f_T} \left\{ \max_{p^* \in \mathcal{S}^n} \sum_{t=1}^{T} f_t(p^* x_t) - \sum_{t=1}^{T} f_t(p_t x_t) \right\} \tag{1}$$

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 Moses Charikar's NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and an Alfred P. Sloan fellowship

†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 $p \in \mathbb{R}^n, \sum_i p_i = 1, p_i \geq 0$

When the f_t's are linear functions, the regret can be as bad as $\Omega(G\sqrt{T})$, where the derivatives of the payoff functions are bounded by $f'_t \leq G$. Algorithms which achieve this regret bounds have been independently discovered by Hannan [Han57], Fudenberg and Levin [FL99], Foster and Vohra, Freund and Schapire [FS97] and many others in the contexts of game theory, statistics, finance, operations research and machine learning. We refer the reader to the excellent surveys by Blum [Blu98] and Foster and Vohra [FV99].

As for the more general framework, an example of a well-studied problem is that 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 $O(n \log T)$ regret. The downside of Cover's algorithm is its running time which is exponential in the dimension, namely $\Omega(T^n)$. The running time was subsequently improved by Kalai and Vempala [KV03] to a large polynomial in n and T using random walks for sampling logconcave functions. It is possible that recent improvements in random sampling may lead to much improved running time for the Kalai-Vempala algorithm, yet even optimistic estimates seem to be slower by a $poly(n)$ factor then our algorithm [Vem06].

In this paper we propose and analyze an algorithm that combine the benefits of previous approaches: both computational efficiency (the running time per iteration is $\tilde{O}(n^3T)$ [2] and $\tilde{O}(n^3)$ if the payoff functions are polynomials) and $\tilde{O}(\frac{G^2}{H} n \log T)$ regret , where as before G is a bound on the first derivative of the payoff functions and the second derivative is bounded by $|f''_t| \geq H > 0$. 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.

Zinkevich [Zin03] provides an algorithm that achieves a weaker bound of $\tilde{O}(G^2 n\sqrt{T})$ regret in a very general framework. His observation is that linear payoff functions are in some sense the "most difficult", and therefore proceeds to approximate curved functions by their linear approximation and achieves the same upper and lower bounds as for game playing with linear payoff functions. In this paper we inherently exploit the curvature (second derivative) of the payoff functions. This additional information enables our algorithm to break the $\Omega(\sqrt{T})$ bound and attain $O(\log T)$ regret (which previously was known only for portfolio management, as stated previously).

As mentioned above, when the payoff functions are linear, the regret is lower bounded by $\Omega(\sqrt{T})$. We also provide a simple deterministic version of our algorithm that attains this regret for linear functions, i.e. $O(G^2 n\sqrt{T})$ regret, hence derandomizing an algorithm by Kalai and Vempala [KV05] with many applications to online optimization.

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. For linear payoff functions, the method was rigourously 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 payoff 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].

[2] we use the $\tilde{O}$ notation to hide terms with polylogarithmic dependence on n,T,G,H

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 that defines the best mixed strategy in hindsight. We show that this variant of follow-the-leader ensures logarithmic regret for strictly concave payoff functions (by which we mean $H > 0$), thereby answering the question above. Smooth Prediction is a *deterministic* algorithm that has running time of $\tilde{O}(n^3T)$, a significant improvement over previous methods. In case the payoff functions are concave polynomials, the running time can be further improved to $\tilde{O}(n^3)$.

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, which is a second order method . By contrast, many prior algorithms are based on the the Multiplicative Weights Updates Method (for a survey see [AHK05], and also section 1.3), which is related to Gradient Descent, a first order method.

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)[3] 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 problem of minimizing the regret is a subcase of our online game playing model where the concave function applied at each iteration is $f_t(px) = \log(px)$.

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, whose running time has already been discussed in detail earlier.

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 in this paper applies to universal portfolio management. Under the "bounded variability" assumption, it is the most efficient algorithm yet to attain optimal $O(\log T)$ regret. The algorithm can be modified such that the regret remains sublinear even without the bounded variability assumption.

[3] 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.

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. In subsequent work, the techniques developed hereby were extended to the full Zinkevich model [HKKA06].

Other applications of our result include deriving approximate optimization algorithms which fall into the category of lagrangian relaxation [PST91, You95, GK98, Fle00]. The improved bounds on regret achieved in this paper imply that the **average regret** (i.e. regret per iteration, which is just the regret as defined in equation (1) divided by the number of iterations T) is bounded by ε after $\tilde{O}(\frac{G^2 n}{H} \cdot \frac{1}{\varepsilon})$ iterations. This is in contrast to previous online algorithms which achieve regret ε after $\Omega(\frac{G}{\varepsilon^2})$ iterations, and leads to more efficient algorithms for a variety of optimization problems [Haz06]. This improvement is reminiscent of, but different than, the recent results of Bienstock and Iyengar [BI04].

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 $f_t''(px_t) \leq -H < 0, \forall t, p$.

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

$$\max_{p \in R^n} \left(\sum_{i=1}^{t-1} f_i(px_i) + \sum_{i \in n} \log(pe_i) \right) \tag{2}$$

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

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

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

$$\textit{Regret}(\text{Smooth Prediction}) \leq 4n \frac{G^2}{H} \log(\omega n T)$$

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

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

In section 4 we prove that even without a bound on the second derivative, a modified version of Smooth Prediction has $O(\sqrt{T})$ regret, which is optimal for linear functions. This implies sublinear regret for portfolio management even without the "bounded variability" assumption.

2.2 Running Time

Interior point methods [NN94] allow maximization of a n-dimensional concave function over the n-dimensional simplex in time $\tilde{O}(n^3)$. 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 $\tilde{O}(T \cdot n^3)$. 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 $\tilde{O}(Tn^3)$. In case the functions f_t are polynomials, we need only save the sum of all coefficients to optimize over the sum of the functions. Therefore, the computation can be carried out in time $\tilde{O}(k \cdot n^3)$ for polynomials with k coefficients.

We note that in practice, many times approximations to p_T are sufficient, such as those obtained by lagrangian relaxation methods. Specifically for portfolio management, one could use the 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 (2) 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

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

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 $\nabla f_t(px_t) = f_t'(px_t)x_t$. From both observations we have

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

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 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:

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

To continue, we use the following lemma, 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 $f_t(p) = f_t(px_t)$, and let $f_0(p) = \sum_{i=1}^{n} \log(pe_i)$ 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

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

Proof: By the definition of p_T, we have:

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

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

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

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

Lemma 6 now follows as a corollary:

Proof:[Lemma 6] Combining the previous two claims:

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

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

4 Near-linear payoff functions

Theorem 1 bounds the regret with an inverse dependence on the second derivative of the payoff functions, and hence provides very poor bounds in case the second derivative of the payoff functions is extremely small and no meaningful bounds at all in the case of linear payoff functions. However, a simple trick can be used to obtain $\tilde{O}(\sqrt{T})$ regret over all choices of payoff functions. Note that this regret is optimal for linear payoff functions. Moreover, the simple modification retains the benefits of Smooth Prediction, namely efficiency and determinism.

The modification consists of adding a small fictitious non-linear term to the payoff function. This has the benefit of bounding the second derivative from below, albeit deviating from the original regret goal. Choosing the tradeoff correctly ensures low regret, as formalized below.

Theorem 7 *For any payoff function $f_t : \mathbb{R} \mapsto \mathbb{R}$ encountered throughout the online game, consider $\tilde{f}_t(px) \triangleq f(px) - \frac{1}{\sqrt{t}}(px)^2$. Applying* Smooth Prediction *to the functions $\tilde{f}_t$ we obtain*

$$\textit{Regret}(\text{Smooth Prediction}) = \tilde{O}(G^2 n\sqrt{T})$$

Proof: The regret of Smooth Prediction on this modified sequence of functions is $\sum_{t=1}^{T}[f_t(p^* x_t) - f_t(p_t x_t)]$ which can be bound by

$$\begin{aligned}
\sum_{t=1}^{T}[f_t(p^*x_t) - f_t(p_tx_t)] &= \sum_{t=1}^{T}[f_t(p^*x_t) - \tilde{f}_t(p^*x_t) + \tilde{f}_t(p^*x_t) - \tilde{f}_t(p_tx_t) + \tilde{f}_t(p_tx_t) - f_t(p_tx_t)] \\
&\leq \sum_{t=1}^{T}[\tilde{f}_t(p^*x_t) - \tilde{f}_t(p_tx_t)] + \sum_{t=1}^{T}\frac{1}{\sqrt{t}}((p^*x_t)^2 - (p_tx_t)^2) \\
&\leq 4n\frac{G^2}{T^{-1/2}}\log(\omega nT) + \sum_{t=1}^{T}\frac{1}{\sqrt{t}}
\end{aligned}$$

Where the last inequality follows from Theorem 1 taking into account that the second derivative of $\tilde{f}_t$ is bounded by

$$\tilde{f}''_t \leq f''_t - \frac{1}{\sqrt{t}} \leq -\frac{1}{\sqrt{T}}$$

This implies the $\tilde{O}(G^2 n\sqrt{T})$ bound on the regret. □

5 Future work

For portfolio management, it would be interesting to extend the framework hereby to take into account transaction costs and/or side information, as achieved by Blum and Kalai [BK97] for Cover's original algorithm.

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 first author would like to thank Moses Charikar for his immense support and constant guidance. 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.

[BI04] Daniel Bienstock and Garud Iyengar. Solving fractional packing problems in $^{\text{ast}}(1/?)$ iterations. In *STOC*, pages 146–155, 2004.

[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.

[Blu98] A. Blum. On-line algorithms in machine learning. In A. Fiat and G. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 306–325. LNCS 1442, 1998.

[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.

[FL99] Drew Fudenberg and David K. Levine. An easier way to calibrate. *Games and Economic Behavior*, 29(1-2):131–137, October 1999. available at http://ideas.repec.org/a/eee/gamebe/v29y1999i1-2p131-137.html.

[Fle00] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.*, 13(4):505–520, 2000.

[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.

[FV99] Dean P. Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1-2):7–35, October 1999. available at http://ideas.repec.org/a/eee/gamebe/v29y1999i1-2p7-35.html.

[GK98] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science(FOCS-98)*, pages 300–309, Los Alamitos, CA, November8–11 1998. IEEE Computer Society.

[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.

[Haz06] Elad Hazan. Approximate convex optimization by online game playing. *Manuscript*, 2006.

[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.

[PST91] Serge A. Plotkin, David B. Shmoys, and Tardos Tardos. Fast approximation algorithm for fractional packing and covering problems. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science, FOCS'91 (San Juan, Puerto Rico, October 1-4, 1991)*, pages 495–504, Los Alamitos-Washington-Brussels-Tokyo, 1991. IEEE Computer Society Press.

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

[Vem06] S. Vempala. Personal communication, 2006.

[You95] Neal E. Young. Randomized rounding without solving the linear program. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 170–178, San Francisco, California, 22–24 January 1995.

[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*

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

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 5 now follows as a corollary:

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

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

□

B Proof of Lemma 4

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 4 follows as a corollary of this claim.