Logarithmic Regret Algorithms for Online Convex Optimization

Elad Hazan[1,⋆], Adam Kalai[2], Satyen Kale[1,⋆], and Amit Agarwal[1]

[1] Princeton University
{ehazan, satyen, aagarwal}@princeton.edu
[2] TTI-Chicago
kalai@tti-c.org

Abstract. In an online convex optimization problem a decision-maker makes a sequence of decisions, i.e., chooses a sequence of points in Euclidean space, from a fixed feasible set. After each point is chosen, it encounters a sequence of (possibly unrelated) convex cost functions. Zinkevich [Zin03] introduced this framework, which models many natural repeated decision-making problems and generalizes many existing problems such as Prediction from Expert Advice and Cover's Universal Portfolios. Zinkevich showed that a simple online gradient descent algorithm achieves additive *regret* $O(\sqrt{T})$, for an arbitrary sequence of T convex cost functions (of bounded gradients), with respect to the best single decision in hindsight.

In this paper, we give algorithms that achieve regret $O(\log(T))$ for an arbitrary sequence of strictly convex functions (with bounded first and second derivatives). This mirrors what has been done for the special cases of prediction from expert advice by Kivinen and Warmuth [KW99], and Universal Portfolios by Cover [Cov91]. We propose several algorithms achieving logarithmic regret, which besides being more general are also much more efficient to implement.

The main new ideas give rise to an efficient algorithm based on the Newton method for optimization, a new tool in the field. Our analysis shows a surprising connection to follow-the-leader method, and builds on the recent work of Agarwal and Hazan [AH05]. We also analyze other algorithms, which tie together several different previous approaches including follow-the-leader, exponential weighting, Cover's algorithm and gradient descent.

1 Introduction

In the problem of *online convex optimization* [Zin03], there is a fixed convex compact feasible set $K \subset \mathbb{R}^n$ and an *arbitrary, unknown* sequence of convex cost functions $f_1, f_2, \ldots : K \to \mathbb{R}$. The decision maker must make a sequence of decisions, where the t^{th} decision is a selection of a point $x_t \in K$ and there is a cost of $f_t(x_t)$ on period t. However, x_t is chosen with only the knowledge of

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

G. Lugosi and H.U. Simon (Eds.): COLT 2006, LNAI 4005, pp. 499–513, 2006.

the set K, previous points $x_1, \ldots, x_{t-1}$, and the previous functions $f_1, \ldots, f_{t-1}$. Examples include many repeated decision-problems:

Example 1: Production. Consider a company deciding how much of n different products to produce. In this case, their profit may be assumed to be a concave function of their production (the goal is maximize profit rather than minimize cost). This decision is made repeatedly, and the model allows the profit functions to be changing arbitrary concave functions, which may depend on various factors such as the economy.

Example 2: Linear prediction with a convex loss function. In this setting, there is a sequence of examples $(p_1, q_1), \ldots, (p_T, q_T) \in \mathbb{R}^n \times [0, 1]$. For each $t = 1, 2, \ldots, T$, the decision-maker makes a *linear* prediction of $q_t \in [0, 1]$ which is $x_t^\top p_t$, for some $x_t \in \mathbb{R}^n$, and suffers some loss $\mathcal{L}(q_t, x_t^\top p_t)$, where $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is some fixed, known convex loss function, such as quadratic $\mathcal{L}(q, q') = (q - q')^2$. The online convex optimization framework permits this example, because the function $f_t(x) = \mathcal{L}(q_t, x^\top p_t)$ is a convex function of $x \in \mathbb{R}^n$. This problem of linear prediction with a convex loss function has been well studied (e.g., [CBL06]), and hence one would prefer to use the near-optimal algorithms that have been developed especially for that problem. We mention this application only to point out the generality of the online convex optimization framework.

Example 3: Portfolio management. In this setting, for each $t = 1, ..., T$ an online investor chooses a distribution x_t over n stocks in the market. The market outcome at iteration t is captured by a price relatives vector c_t, such that the loss to the investor is $-\log(x_t^\top c_t)$ (see Cover [Cov91] for motivation and more detail regarding the model). Again, the online convex optimization framework permits this example, because the function $f_t(x) = -\log(x^\top c)$ is a convex function of $x \in \mathbb{R}^n$.

This paper shows how three seemingly different approaches can be used to achieve logarithmic regret in the case of some higher-order derivative assumptions on the functions. The algorithms are relatively easy to state. In some cases, the analysis is simple, and in others it relies on a carefully constructed potential function due to Agarwal and Hazan [AH05]. Lastly, our gradient descent results relate to previous analyses of stochastic gradient descent [Spa03], which is known to converge at a rate of $1/T$ for T steps of gradient descent under various assumptions on the distribution over functions. Our results imply a $\log(T)/T$ convergence rate for the same problems, though as common in the online setting, the assumptions and guarantees are simpler and stronger than their stochastic counterparts.

1.1 Our Results

The *regret* of the decision maker at time T is defined to be its total cost minus the cost of the best single decision, where the best is chosen with the benefit of hindsight.

$$\text{regret}_T = \text{regret} = \textstyle\sum_{t=1}^T f_t(x_t) - \min_{x \in K} \sum_{t=1}^T f_t(x).$$

A standard goal in machine learning and game theory is to achieve algorithms with guaranteed low regret (this goal is also motivated by psychology). Zinkevich showed that one can guarantee $O(\sqrt{T})$ regret for an *arbitrary* sequence of differentiable convex functions of bounded gradient, which is tight up to constant factors. In fact, $\Omega(\sqrt{T})$ regret is unavoidable even when the functions come from a fixed distribution rather than being chosen adversarially. [1]

Variable	Meaning
$K \subseteq \mathbb{R}^n$	the convex compact feasible set
$D \geq 0$	the diameter of K, $D = \sup_{x,y \in K} \\|x - y\\|$
$f_1, \ldots, f_T$	Sequence of T twice-differentiable convex functions $f_t : \mathbb{R}^n \to \mathbb{R}$.
$G \geq 0$	$\\|\nabla f_t(x)\\| \leq G$ for all $x \in K, t \leq T$ (in one dimension, $\|f_t'(x)\| \leq G$.)
$H \geq 0$	$\nabla^2 f_t(x) \succeq H I_n$ for all $x \in K, t \leq T$ (in one dimension, $f_t''(x) \geq H$).
$\alpha \geq 0$	Such that $\exp(-\alpha f_t(x))$ is a concave function of $x \in K$, for $t \leq T$.

Fig. 1. Notation in the paper. Arbitrary convex functions are allowed for $G = \infty, H = 0, \alpha = 0$. $\\|\cdot\\|$ is the ℓ_2 (Euclidean) norm.

Algorithm	Regret bound
Online gradient descent	$\frac{G^2}{2H}(1 + \log T)$
Online Newton step	$3(\frac{1}{\alpha} + 4GD)n \log T$
Exponentially weighted online opt.	$\frac{n}{\alpha}(1 + \log(1 + T))$

Fig. 2. Results from this paper. Zinkevich achieves $GD\sqrt{T}$, even for $H = \alpha = 0$.

Our notation and results are summarized in Figures 1 and 2. Throughout the paper we denote by $\|\cdot\|$ the ℓ_2 (Euclidean) norm. We show $O(\log T)$ regret under relatively weak assumptions on the functions $f_1, f_2, \ldots$. Natural assumptions to consider might be that the gradients of each function are of bounded magnitude G, i.e., $\|\nabla f_t(x)\| \leq G$ for all $x \in K$, and that each function in the sequence is *strongly-concave*, meaning that the second derivative is bounded away from 0. In one dimension, these assumptions correspond simply to $|f_t'(x)| \leq G$ and $f_t''(x) \geq H$ for some $G, H > 0$. In higher dimensions, one may require these properties to hold on the functions in *every direction* (i.e., for the 1-dimensional function of θ, $f_t(\theta u)$, for any unit vector $u \in \mathbb{R}^n$), which can be equivalently written in the following similar form: $\|\nabla f_t(x)\| \leq G$ and $\nabla^2 f_t(x) \succeq H I_n$, where I_n is the $n \times n$ identity matrix and we write $A \succeq B$ if the matrix $A - B$ is positive semi-definite (symmetric with non-negative eigenvalues).

Intuitively, it is easier to minimize functions that are "very concave," and the above assumptions may seem innocuous enough. However, they rule out several interesting types of functions. For example, consider the function $f(x) =$

[1] This can be seen by a simple randomized example. Consider $K = [-1, 1]$ and linear functions $f_t(x) = r_t x$, where $r_t = \pm 1$ are chosen in advance, independently with equal probability. $\mathbf{E}_{r_t}[f_t(x_t)] = 0$ for any t and x_t chosen online, by independence of x_t and r_t. However, $\mathbf{E}_{r_1,\ldots,r_T}[\min_{x \in K} \sum_1^T f_t(x)] = \mathbf{E}[-|\sum_1^T r_t|] = -\Omega(\sqrt{T})$.

$(x^\top w)^2$, for some vector $w \in \mathbb{R}^n$. This is strongly convex in the direction w, but is constant in directions orthogonal to w. A simpler example is the constant function $f(x) = c$ which is not strongly convex, yet is easily (and unavoidably) minimized.

Some of our algorithms also work without explicitly requiring $H > 0$, i.e., when $G = \infty, H = 0$. In these cases we require that there exists some $\alpha > 0$ such that $h_t(x) = \exp(-\alpha f_t(x))$ is a concave function of $x \in K$, for all t. A similar *exp-concave* assumption has been utilized for the prediction for expert-advice problem [CBL06]. It turns out that given the bounds G and H, the exp-concavity assumption holds with $\alpha = G^2/H$. To see this in one dimension, one can easily verify the assumption on one-dimensional functions $f_t : \mathbb{R} \to \mathbb{R}$ by taking two derivatives,

$$h_t''(x) = ((\alpha f_t'(x))^2 - \alpha f_t''(x)) \exp(-\alpha f_t(x)) \leq 0 \iff \alpha \leq \frac{f_t''(x)}{(f_t'(x))^2}.$$

All of our conditions hold in n-dimensions if they hold in every direction. Hence we have that the exp-concave assumption is a *weaker* assumption than those of G, H, for $\alpha = G^2/H$. This enables us to compare the three regret bounds of Figure 2. In these terms, Online Gradient Descent requires the strongest assumptions, whereas Exponentially Weighted Online Optimization requires only exp-concavity (and not even a bound on the gradient). Perhaps most interesting is Online Newton Step which requires relatively weak assumptions and yet, as we shall see, is very efficient to implement (and whose analysis is the most technical).

2 The Algorithms

The algorithms are presented in Figure 3. The intuition behind most of our algorithms stem from new observations regarding the well studied *follow-the-leader* method (see [Han57, KV05, AH05]).

The basic method, which by itself fails to provide sub-linear regret let alone logarithmic regret, simply chooses on period t the single fixed decision that would have been the best to use on the previous $t-1$ periods. This corresponds to choosing $x_t = \arg\min_{x \in K} \sum_{\tau=1}^{t-1} f_\tau(x)$. The standard approach to analyze such algorithms proceeds by inductively showing,

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} f_t(x_t) - f_t(x_{t+1}) \quad (1)$$

The standard analysis proceeds by showing that the leader doesn't change too much, i.e. $x_t \approx x_{t+1}$, which in turn implies low regret.

One of the significant deviations from this standard analysis is in the variant of follow-the-leader called Online Newton Step. The analysis does not follow this paradigm directly, but rather shows *average stability* (i.e. that $x_t \approx x_{t+1}$

ONLINE GRADIENT DESCENT. (Zinkevich's online version of Stochastic Gradient Descent)
Inputs: convex set $K \subset \mathbb{R}^n$, step sizes $\eta_1, \eta_2, \ldots \geq 0$.

- On period 1, play an arbitrary $x_1 \in K$.
- On period $t > 1$: play

$$x_t = \Pi_K(x_{t-1} - \eta_t \nabla f_{t-1}(x_{t-1}))$$

Here, Π_K denotes the *projection* onto nearest point in K, $\Pi_K(y) = \arg\min_{x \in K} \|x - y\|$.

ONLINE NEWTON STEP.
Inputs: convex set $K \subset \mathbb{R}^n$, and the parameter β.

- On period 1, play an arbitrary $x_1 \in K$.
- On period $t > 1$: play the point x_t given by the following equations:

$$\begin{aligned} \nabla_{t-1} &= \nabla f_{t-1}(x_{t-1}) \\ A_{t-1} &= \sum_{\tau=1}^{t-1} \nabla_\tau \nabla_\tau^\top \\ b_{t-1} &= \sum_{\tau=1}^{t-1} \nabla_\tau \nabla_\tau^\top x_\tau - \frac{1}{\beta} \nabla_\tau \\ x_t &= \Pi_K^{A_{t-1}} \left(A_{t-1}^{-1} b_{t-1} \right) \end{aligned}$$

Here, $\Pi_K^{A_{t-1}}$ is the projection in the norm induced by A_{t-1}:

$$\Pi_K^{A_{t-1}}(y) = \arg\min_{x \in K} \, (x - y)^\top A_{t-1}(x - y)$$

A_{t-1}^{-1} denotes the Moore-Penrose pseudoinverse of A_{t-1}.

EXPONENTIALLY WEIGHTED ONLINE OPTIMIZATION.
Inputs: convex set $K \subset \mathbb{R}^n$, and the parameter α.

- Define weights $w_t(x) = \exp(-\alpha \sum_{\tau=1}^{t-1} f_\tau(x))$.
- On period t play $x_t = \frac{\int_K x w_t(x) dx}{\int_K w_t(x) dx}$.
  (*Remark:* choosing x_t at random with density proportional to $w_t(x)$ also gives our bounds.)

Fig. 3. Online optimization algorithms

on the "average", rather than always) using an extension of the Agarwal-Hazan potential function.

Another building block, due to Zinkevich [Zin03], is that if we have another set of functions $\tilde{f}_t$ for which $\tilde{f}_t(x_t) = f_t(x_t)$ and $\tilde{f}_t$ is a lower-bound on f_t, so $\tilde{f}_t(x) \leq f_t(x)$ for all $x \in K$, then it suffices to bound the regret with respect to $\tilde{f}_t$, because,

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \le \sum_{t=1}^{T} \tilde{f}_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} \tilde{f}_t(x) \quad (2)$$

He uses this observation in conjunction with the fact that a convex function is lower-bounded by its tangent hyperplanes, to argue that it suffices to analyze online gradient descent for the case of linear functions.

We observe[2] that online gradient descent can be viewed as running follow-the-leader on the sequence of functions $\tilde{f}_0(x) = (x - x_1)^2/\eta$ and $\tilde{f}_t(x) = f_t(x_t) + \nabla f_t(x_t)^\top (x - x_t)$. To do this, one need only calculate the minimum of $\sum_{\tau=0}^{t-1} \tilde{f}_\tau(x)$.

As explained before, any algorithm for the online convex optimization problem with linear functions has $\Omega(\sqrt{T})$ regret, and thus to achieve logarithmic regret one necessarily needs to use the curvature of functions. When we consider strongly concave functions where $H > 0$, we can lower-bound the function f_t by a paraboloid,

$$f_t(x) \ge f_t(x_t) + \nabla f_t(x_t)^\top (x - x_t) + \frac{H}{2}(x - x_t)^2,$$

rather than a linear function. The follow-the-leader calculation, however, remains similar. The only difference is that the step-size $\eta_t = 1/(Ht)$ decreases linearly rather than as $O(1/\sqrt{t})$.

For functions which permit $\alpha > 0$ such that $\exp(-\alpha f_t(x))$ is concave, it turns out that they can be lower-bounded by a paraboloid $\tilde{f}_t(x) = a + (w^\top x - b)^2$ where $w \in \mathbb{R}^n$ is proportional to $\nabla f_t(x_t)$ and $a, b \in \mathbb{R}$. Hence, one can do a similar follow-the-leader calculation, and this gives the Follow The Approximate Leader algorithm in Figure 4. Formally, the Online Newton Step algorithm is an efficient implementation to the follow-the-leader variant Follow The Approximate Leader (see Lemma 3), and clearly demonstrates its close connection to the Newton method from classical optimization theory. Interestingly, the derived Online Newton Step algorithm does not directly use the Hessians of the observed functions, but only a lower-bound on the Hessians, which can be calculated from the $\alpha > 0$ bound.

Finally, our Exponentially Weighted Online Optimization algorithm does not seem to be directly related to follow-the-leader. It is more related to similar algorithms which are used in the problem of prediction from expert advice[3] and to Cover's algorithm for universal portfolio management.

2.1 Implementation and Running Time

Perhaps the main contribution of this paper is the introduction of a general logarithmic regret algorithms that are efficient and relatively easy to implement. The algorithms in Figure 3 are described in their mathematically simplest forms, but

[2] Kakade has made a similar observation [Kak05].

[3] The standard weighted majority algorithm can be viewed as picking an expert of minimal cost when an additional random cost of $-\frac{1}{\eta} \ln \ln r_i$ is added to each expert, where r_i is chosen independently from $[0, 1]$.

Follow The Approximate Leader.
Inputs: convex set $K \subset \mathbb{R}^n$, and the parameter β.

- On period 1, play an arbitrary $x_1 \in K$.
- On period t, play the leader x_t defined as

$$x_t \triangleq \arg\min_{x \in K} \sum_{\tau=1}^{t-1} \tilde{f}_\tau(x)$$

Where for $\tau = 1, \ldots, t-1$, define $\nabla_\tau = \nabla f_\tau(x_\tau)$ and

$$\tilde{f}_\tau(x) \triangleq f_\tau(x_\tau) + \nabla_\tau^\top (x - x_\tau) + \frac{\beta}{2}(x - x_\tau)^\top \nabla_\tau \nabla_\tau^\top (x - x_\tau)$$

Fig. 4. The Follow The Approximate Leader algorithm, which is equivalent to Online Newton Step

implementation has been disregarded. In this section, we discuss implementation issues and compare the running time of the different algorithms.

The Online Gradient Descent algorithm is straightforward to implement, and updates take time $O(n)$ given the gradient. However, there is a projection step which may take longer. For many convex sets such as a ball, cube, or simplex, computing Π_K is fast and straightforward. For convex polytopes, the projection oracle can be implemented efficiently using interior point methods. In general, K can be specified by a membership oracle ($\chi_K(x) = 1$ if $x \in K$ and 0 if $x \notin K$), along with a point $x_0 \in K$ as well as radii $R \geq r > 0$ such that the balls of radii R and r around x_0 contain and are contained in K, respectively. In this case Π_K can be computed (to ε accuracy) in time $\tilde{O}(n^4 \log(\frac{R}{r}))$ [4] using the Vaidya's algorithm [Vai96].

The Online Newton Step algorithm requires $O(n^2)$ space to store the matrix A_t. Every iteration requires the computation of the matrix A_t^{-1}, the current gradient, a matrix-vector product and possibly a projection onto K.

A naïve implementation would require computing the Moore-Penrose pseudoinverse of the matrix A_t every iteration. However, in case A_t is invertible, the matrix inversion lemma [Bro05] states that for invertible matrix A and vector x

$$(A + xx^\top)^{-1} = A^{-1} - \frac{A^{-1}xx^\top A^{-1}}{1 + x^\top A^{-1}x}$$

Thus, given A_{t-1}^{-1} and ∇_t one can compute A_t^{-1} in time $O(n^2)$. A generalized matrix inversion lemma [Rie91] allows for iterative update of the pseudoinverse also in time $O(n^2)$, details will appear in the full version.

The Online Newton Step algorithm also needs to make projections onto K, but of a slightly different nature than Online Gradient Descent. The required projection, denoted by $\Pi_K^{A_t}$, is in the vector norm induced by the matrix A_t, viz. $\|x\|_{A_t} = \sqrt{x^\top A_t x}$. It is equivalent to finding the point $x \in K$ which

[4] The $\tilde{O}$ notation hides poly-logarithmic factors, in this case $\log(nT/\varepsilon)$.

minimizes $(x-y)^\top A_t(x-y)$ where y is the point we are projecting. We assume the existence of an oracle which implements such a projection given y and A_t. The runtime is similar to that of the projection step of Online Gradient Descent.

Modulo calls to the projections oracle, the Online Newton Step algorithm can be implemented in time and space $O(n^2)$, requiring only the gradient at each step.

The Exponentially Weighted Online Optimization algorithm can be approximated by sampling points according to the distribution with density proportional to w_t and then taking their mean. In fact, as far as an expected guarantee is concerned, our analysis actually shows that the algorithm which chooses a single random point x_t with density proportional to $w_t(x)$ achieves the stated regret bound, in expectation. Using recent random walk analyses of Lovász and Vempala [LV03a, LV03b], m samples from such a distribution can be computed in time $\tilde{O}((n^4+mn^3)\log\frac{R}{r})$. A similar application of random walks was used previously for an efficient implementation of Cover's Universal Portfolio algorithm [KV03].

3 Analysis

3.1 Online Gradient Descent

Theorem 1. *Assume that the functions f_t have bounded gradient, $\|\nabla f_t(x)\| \leq G$, and Hessian, $\nabla^2 f_t(x) \succeq HI_n$, for all $x \in K$.*

The Online Gradient Descent *algorithm of Figure 3, with $\eta_t = (Ht)^{-1}$ achieves the following guarantee, for all $T \geq 1$.*

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in K}\sum_{t=1}^{T} f_t(x) \;\leq\; \frac{G^2}{2H}(1+\log T)$$

Proof. Let $x^* \in \arg\min_{x\in K}\sum_{t=1}^T f_t(x)$. Define $\nabla_t \triangleq \nabla f_t(x_t)$. By H-strong convexity, we have,

$$f_t(x^*) \geq f_t(x_t) + \nabla_t^\top(x^*-x_t) + \frac{H}{2}\|x^*-x_t\|^2$$
$$2(f_t(x_t)-f_t(x^*)) \leq 2\nabla_t^\top(x_t-x^*) - H\|x^*-x_t\|^2 \tag{3}$$

Following Zinkevich's analysis, we upper-bound $\nabla_t^\top(x_t-x^*)$. Using the update rule for x_{t+1}, we get

$$\|x_{t+1}-x^*\|^2 \;=\; \|\Pi(x_t-\eta_{t+1}\nabla_t)-x^*\|^2 \leq \|x_t-\eta_{t+1}\nabla_t-x^*\|^2.$$

The inequality above follows from the properties of projection onto convex sets. Hence,

$$\|x_{t+1}-x^*\|^2 \;\leq\; \|x_t-x^*\|^2 + \eta_{t+1}^2\|\nabla_t\|^2 - 2\eta_{t+1}\nabla_t^\top(x_t-x^*)$$
$$2\nabla_t^\top(x_t-x^*) \;\leq\; \frac{\|x_t-x^*\|^2-\|x_{t+1}-x^*\|^2}{\eta_{t+1}} + \eta_{t+1}G^2 \tag{4}$$

Sum up (4) from $t = 1$ to T. Set $\eta_{t+1} = 1/(Ht)$, and using (3), we have:

$$\begin{aligned} 2\sum_{t=1}^{T} f_t(x_t) - f_t(x^*) &\le \sum_{t=1}^{T} \|x_t - x^*\|^2 \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} - H\right) + G^2 \sum_{t=1}^{T} \eta_{t+1} \\ &= 0 + G^2 \sum_{t=1}^{T} \frac{1}{Ht} \;\le\; \frac{G^2}{H}(1 + \log T) \end{aligned}$$

■

3.2 Online Newton Step

Before analyzing the algorithm, we need a couple of lemmas.

Lemma 2. *If a function $f : K \to \mathbb{R}$ is such that $\exp(-\alpha f(x))$ is concave, and has gradient bounded by $\|\nabla f\| \le G$, then for $\beta = \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$ the following holds:*

$$\forall x, y \in K: \;\; f(x) \ge f(y) + \nabla f(y)^\top (x - y) + \frac{\beta}{2}(x - y)\nabla f(y) \nabla f(y)^\top (x - y)$$

Proof. First, by computing derivatives, one can check that since $\exp(-\alpha f(x))$ is concave and $2\beta \le \alpha$, the function $h(x) = \exp(-2\beta f(x))$ is also concave. Then by the concavity of $h(x)$, we have

$$h(x) \;\le\; h(y) + \nabla h(y)^\top (x - y).$$

Plugging in $\nabla h(y) = -2\beta \exp(-2\beta f(y)) \nabla f(y)$ gives,

$$\exp(-2\beta f(x)) \;\le\; \exp(-2\beta f(y))[1 - 2\beta \nabla f(y)^\top (x - y)].$$

Simplifying

$$f(x) \ge f(y) - \frac{1}{2\beta} \log[1 - 2\beta \nabla f(y)^\top (x - y)].$$

Next, note that $|2\beta \nabla f(y)^\top (x - y)| \le 2\beta GD \le \frac{1}{4}$ and that for $|z| \le \frac{1}{4}$, $-\log(1 - z) \ge z + \frac{1}{4}z^2$. Applying the inequality for $z = 2\beta \nabla f(y)^\top (x - y)$ implies the lemma. ■

Lemma 3. *The* Online Newton Step *algorithm is equivalent to the* Follow The Approximate Leader *algorithm.*

Proof. In the Follow The Approximate Leader algorithm, one needs to perform the following optimization at period t:

$$x_t \;\triangleq\; \arg\min_{x \in K} \sum_{\tau=1}^{t-1} \tilde{f}_\tau(x)$$

By expanding out the expressions for $\tilde{f}_\tau(x)$, multiplying by $\frac{2}{\beta}$ and getting rid of constants, the problem reduces to minimizing the following function over $x \in K$:

$$\sum_{\tau=1}^{t-1} x^\top \nabla_\tau \nabla_\tau^\top x - 2(x_\tau^\top \nabla_\tau \nabla_\tau^\top - \frac{1}{\beta}\nabla_\tau^\top)x$$
$$= x^\top A_{t-1} x - 2b_{t-1}^\top x = (x - A_{t-1}^{-1} b_{t-1})^\top A_{t-1}(x - A_{t-1}^{-1} b_{t-1}) - b_{t-1}^\top A_{t-1}^{-1} b_{t-1}$$

The solution of this minimization is exactly the projection $\Pi_K^{A_{t-1}}(A_{t-1}^{-1} b_{t-1})$ as specified by Online Newton Step. ∎

Theorem 4. *Assume that the functions* f_t *are such that* $\exp(-\alpha f_t(x))$ *is concave and have gradients bounded by* $\|\nabla f_t(x)\| \leq G$. *Then the* Online Newton Step *algorithm with parameter* $\beta = \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$ *achieves the following guarantee, for all* $T \geq 1$.

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \leq 3\left[\frac{1}{\alpha} + 4GD\right] n \log T$$

Proof. The theorem relies on the observation that by Lemma 2, the function $\tilde{f}_t(x)$ defined by the Follow The Approximate Leader algorithm satisfies $\tilde{f}_t(x_t) = f_t(x_t)$ and $\tilde{f}_t(x) \leq f_t(x)$ for all $x \in K$. Then the inequality (2) implies that it suffices to show a regret bound for the follow-the-leader algorithm run on the $\tilde{f}_t$ functions. The inequality (1) implies that it suffices to bound $\sum_{t=1}^{T}\left[\tilde{f}_t(x_t) - \tilde{f}_t(x_{t+1})\right]$, which is done in Lemma 5 below. ∎

Lemma 5.

$$\sum_{t=1}^{T}\left[\tilde{f}_t(x_t) - \tilde{f}_t(x_{t+1})\right] \leq 3\left[\frac{1}{\alpha} + 4GD\right] n \log T$$

Proof (Lemma 5). For the sake of readability, we introduce some notation. Define the function $F_t \triangleq \sum_{\tau=1}^{t-1} \tilde{f}_\tau$. Note that $\nabla f_t(x_t) = \nabla \tilde{f}_t(x_t)$ by the definition of $\tilde{f}_t$, so we will use the same notation ∇_t to refer to both. Finally, let Δ be the forward difference operator, for example, $\Delta x_t = (x_{t+1} - x_t)$ and $\Delta \nabla F_t(x_t) = (\nabla F_{t+1}(x_{t+1}) - \nabla F_t(x_t))$.

We use the gradient bound, which follows from the convexity of $\tilde{f}_t$:

$$\tilde{f}_t(x_t) - \tilde{f}_t(x_{t+1}) \leq -\nabla \tilde{f}_t(x_t)^\top (x_{t+1} - x_t) = -\nabla_t^\top \Delta x_t \tag{5}$$

The gradient of F_{t+1} can be written as:

$$\nabla F_{t+1}(x) = \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau) + \beta \nabla f_\tau(x_\tau) \nabla f_\tau(x_\tau)^\top (x - x_\tau) \tag{6}$$

Therefore,

$$\nabla F_{t+1}(x_{t+1}) - \nabla F_{t+1}(x_t) = \beta \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau) \nabla f_\tau(x_\tau)^\top \Delta x_t = \beta A_t \Delta x_t \tag{7}$$

The LHS of (7) is

$$\nabla F_{t+1}(x_{t+1}) - \nabla F_{t+1}(x_t) \;=\; \Delta\nabla F_t(x_t) - \nabla_t \tag{8}$$

Putting (7) and (8) together, and adding $\varepsilon\beta\Delta x_t$ we get

$$\beta(A_t + \varepsilon I_n)\Delta x_t = \Delta\nabla F_t(x_t) - \nabla_t + \varepsilon\beta\Delta x_t \tag{9}$$

Pre-multiplying by $-\frac{1}{\beta}\nabla_t^\top(A_t + \varepsilon I_n)^{-1}$, we get an expression for the gradient bound (5):

$$\begin{aligned} -\nabla_t^\top \Delta x_t \;&=\; -\frac{1}{\beta}\nabla_t^\top(A_t + \varepsilon I_n)^{-1}[\Delta\nabla F_t(x_t) - \nabla_t + \varepsilon\beta\Delta x_t] \\ &=\; -\frac{1}{\beta}\nabla_t^\top(A_t + \varepsilon I_n)^{-1}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t] + \frac{1}{\beta}\nabla_t^\top(A_t + \varepsilon I_n)^{-1}\nabla_t \end{aligned} \tag{10}$$

Claim. The first term of (10) can be bounded as follows:

$$-\frac{1}{\beta}\nabla_t^\top(A_t + \varepsilon I_n)^{-1}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t] \;\le\; \varepsilon\beta D^2$$

Proof. Since x_τ minimizes F_τ over K, we have

$$\nabla F_\tau(x_\tau)^\top(x - x_\tau) \;\ge\; 0 \tag{11}$$

for any point $x \in K$. Using (11) for $\tau = t$ and $\tau = t+1$, we get

$$0 \;\le\; \nabla F_{t+1}(x_{t+1})^\top(x_t - x_{t+1}) + \nabla F_t(x_t)^\top(x_{t+1} - x_t) \;=\; -[\Delta\nabla F_t(x_t)]^\top\Delta x_t$$

Reversing the inequality and adding $\varepsilon\beta\|\Delta x_t\|^2 = \varepsilon\beta\Delta x_t^\top\Delta x_t$, we get

$$\begin{aligned} \varepsilon\beta\|\Delta x_t\|^2 \;&\ge\; [\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t]^\top\Delta x_t \\ &=\; \frac{1}{\beta}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t]^\top(A_t + \varepsilon I_n)^{-1}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t - \nabla_t] \\ &\qquad\qquad \text{(by solving for } \Delta x_t \text{ in (9))} \\ &=\; \frac{1}{\beta}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t]^\top(A_t + \varepsilon I_n)^{-1}(\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t) \\ &-\; \frac{1}{\beta}[\Delta\nabla F_t(x_t) + \varepsilon\Delta x_t]^\top(A_t + \varepsilon I_n)^{-1}\nabla_t \\ &\ge\; -\frac{1}{\beta}[\Delta\nabla F_t(x_t) + \varepsilon\beta\Delta x_t]^\top(A_t + \varepsilon I_n)^{-1}\nabla_t \\ &\qquad\qquad \text{(since } (A_t + \varepsilon I_n)^{-1} \succeq 0 \Rightarrow \forall x:\; x^\top(A_t + \varepsilon I_n)^{-1}x \ge 0) \end{aligned}$$

Finally, since the diameter of K is D, we have $\varepsilon\beta\|\Delta x_t\|^2 \le \varepsilon\beta D^2$. ■

Now we bound the second term of (10). Sum up from $t = 1$ to T, and apply Lemma 6 below with $A_0 = \varepsilon I_n$ and $v_t = \nabla_t$. Set $\varepsilon = \frac{1}{\beta^2 D^2 T}$.

$$\frac{1}{\beta}\sum_{t=1}^{T}\nabla_t^\top(A_t+\varepsilon I_n)^{-1}\nabla_t \;\le\; \frac{1}{\beta}\log\left[\frac{|A_T+\varepsilon I_n|}{|\varepsilon I_n|}\right] \le\; \frac{1}{\beta}n\log(\beta^2G^2D^2T^2+1) \;\le\; \frac{2}{\beta}n\log T$$

The second inequality follows since $A_T = \sum_{t=1}^T \nabla_t\nabla_t^\top$ and $\|\nabla_t\| \le G$, we have $|A_T+\varepsilon I_n| \le (G^2T+\varepsilon)^n$.

Combining this inequality with the bound of the claim above, we get

$$\sum_{t=1}^{T}\left[\tilde{f}_t(x_t)-f_t(x_{t+1})\right] \;\le\; \frac{2}{\beta}n\log T+\varepsilon\beta D^2T \;\le\; 3\left[\frac{1}{\alpha}+4GD\right]n\log T$$

as required. ∎

Lemma 6. *Let A_0 be a positive definite matrix, and for $t\ge 1$, let $A_t=\sum_{\tau=1}^t v_tv_t^\top$ for some vectors $v_1,v_2,\ldots,v_t$. Then the following inequality holds:*

$$\sum_{t=1}^{T}v_t^\top(A_t+A_0)^{-1}v_t \;\le\; \log\left[\frac{|A_T+A_0|}{|A_0|}\right]$$

To prove this Lemma, we first require the following claim.

Claim. Let A be a positive definite matrix and x a vector such that $A-xx^\top \succ 0$. Then

$$x^\top A^{-1}x \;\le\; \log\left[\frac{|A|}{|A-xx^\top|}\right]$$

Proof. Let $B=A-xx^\top$. For any positive definite matrix C, let $\lambda_1(C),\lambda_2(C),\ldots,\lambda_n(C)$ be its (positive) eigenvalues.

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

∎

Lemma 6 now follows as a corollary:

Proof (Lemma 6). By the previous claim, we have

$$\sum_{t=1}^{T} v_t^\top (A_t + A_0)^{-1} v_t \;\le\; \sum_{t=1}^{T} \log\left[\frac{|A_t + A_0|}{|A_t + A_0 - v_t v_t^\top|}\right]$$
$$= \sum_{t=2}^{T} \log\left[\frac{|A_t + A_0|}{|A_{t-1} + A_0|}\right] + \log\left[\frac{|A_1 + A_0|}{|A_0|}\right] \;=\; \log\left[\frac{|A_t + A_0|}{|A_0|}\right]$$

■

3.3 Exponentially Weighted Online Optimization

Theorem 7. *Assume that the functions f_t are such that* $\exp(-\alpha f_t(x))$ *is concave. Then the* Exponentially Weighted Online Optimization *algorithm achieves the following guarantee, for all $T \ge 1$.*

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in K} \sum_{t=1}^{T} f_t(x) \le \frac{1}{\alpha} n(1 + \log(1+T)).$$

Proof. Let $h_t(x) = e^{-\alpha f_t(x)}$. The algorithm can be viewed as taking a weighted average over points $x \in K$. Hence, by concavity of h_t,

$$h_t(x_t) \ge \frac{\int_K h_t(x) \prod_{\tau=1}^{t-1} h_\tau(x)\ dx}{\int_K \prod_{\tau=1}^{t-1} h_\tau(x)\ dx}.$$

Hence, we have by telescoping product,

$$\prod_{\tau=1}^{t} h_\tau(x_\tau) \ge \frac{\int_K \prod_{\tau=1}^{t} h_\tau(x)\ dx}{\int_K 1\ dx} = \frac{\int_K \prod_{\tau=1}^{t} h_\tau(x)\ dx}{\mathrm{vol}(K)} \tag{12}$$

Let $x^* = \arg\min_{x \in K} \sum_{t=1}^{T} f_t(x) = \arg\max_{x \in K} \prod_{t=1}^{T} h_t(x)$. Following [BK97], define nearby points $S \subset K$ by,

$$S = \{x \in S | x = \frac{T}{T+1} x^* + \frac{1}{T+1} y, y \in K\}.$$

By concavity of h_t and the fact that h_t is non-negative, we have that,

$$\forall x \in S \quad h_t(x) \ge \frac{T}{T+1} h_t(x^*).$$

Hence,

$$\forall x \in S \quad \prod_{\tau=1}^{T} h_\tau(x) \ge \left(\frac{T}{T+1}\right)^T \prod_{\tau=1}^{T} h_\tau(x^*) \ge \frac{1}{e} \prod_{\tau=1}^{T} h_\tau(x^*)$$

Finally, since $S = x^* + \frac{1}{T+1}K$ is simply a rescaling of K by a factor of $1/(T+1)$ (followed by a translation), and we are in n dimensions, $\mathrm{vol}(S) = \mathrm{vol}(K)/(T+1)^n$. Putting this together with equation (12), we have

$$\prod_{\tau=1}^{T} h_\tau(x_\tau) \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(K)} \frac{1}{e} \prod_{\tau=1}^{T} h_\tau(x^*) \geq \frac{1}{e(T+1)^n} \prod_{\tau=1}^{T} h_\tau(x^*).$$

This implies the theorem. ■

4 Conclusions and Future Work

In this work, we presented efficient algorithms which guarantee logarithmic regret when the loss functions satisfy a mildly restrictive convexity condition. Our algorithms use the very natural follow-the-leader methodology which has been quite useful in other settings, and the efficient implementation of the algorithm shows the connection with the Newton method from offline optimization theory.

Future work involves adapting these algorithms to work in the *bandit* setting, where only the cost of the chosen point is revealed at every point (and no other information). The techniques of Flaxman, Kalai and McMahan [FKM05] seem to be promising for this.

Another direction for future work relies on the observation that the original algorithm of Agarwal and Hazan worked for functions which could be written as a one-dimensional convex function applied to an inner product. However, the analysis requires a stronger condition than the exp-concavity condition we have here. It seems that the original analysis can be made to work just with exp-concavity assumptions, more detail to appear in the full version of this paper.

Acknowledgements

The Princeton authors would like to thank Sanjeev Arora and Rob Schapire for helpful comments.

References

[AH05] Amit Agarwal and Elad Hazan. Efficient algorithms for online game playing and universal portfolio management. *ECCC, TR06-033*, 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.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, Cambridge, 2006.

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

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

[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.

[Kak05] S. Kakade. Personal communication, 2005.

[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.

[KW99] J. Kivinen and M. K. Warmuth. Averaging expert predictions. In *Computational Learning Theory: 4th European Conference (EuroCOLT '99)*, pages 153–167, Berlin, 1999. Springer.

[LV03a] László Lovász and Santosh Vempala. The geometry of logconcave functions and an $o^*(n^3)$ sampling algorithm. Technical Report MSR-TR-2003-04, Microsoft Research, 2003.

[LV03b] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $0^*(n^4)$ volume algorithm. In *Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS)*, pages 650–659, 2003.

[Rie91] Kurt Riedel. A sherman-morrison-woodbury identity for rank augmenting matrices with application to centering. *SIAM J. Mat. Anal.*, 12(1):80–95, January 1991.

[Spa03] J. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc, New York, NY, 2003.

[Vai96] Pravin M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Math. Program.*, 73(3):291–341, 1996.

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