A survey:
The convex optimization approach to regret minimization

Elad Hazan

September 10, 2009

WORKING DRAFT

Abstract

A well studied and general setting for prediction and decision making is regret minimization in games. Originating independently in several disciplines, algorithms for regret minimization have proven to be empirically successful for a wide range of applications.

Recently the design of algorithms for regret minimization in a wide array of settings has been influenced by tools from convex optimization. In this survey we describe two general methods for deriving algorithms and analyzing them, with a "convex optimization flavor". The methods we describe are general enough to capture most existing algorithms with single, simple and generic analysis, and lie at the heart of several recent advancements in prediction theory.

Keywords: online learning, regret, convex optimization

Contents

List of Tables

1 Introduction

In the online decision making scenario, a player has to choose from a pool of available decisions and then occurs a loss corresponding to the quality of decision made. In online regret minimization, the goal is to occur as little loss as the best fixed decision in hindsight. This simple setting has proven to give rise to extremely useful algorithms in practice, as well as a theoretical tool.

Recently tools from convex optimization have infiltrated the field and gave rise to algorithms which are more general, unifying previous results, and many times giving new and improved regret bounds.

The purpose of this survey is to describe two general templates for producing algorithms and proving regret bounds. The templates are very simple, and unify the analysis of many previous algorithms (i.e. multiplicative weights and gradient descent). For the setting of online linear optimization, we also prove that the two templates are equivalent.

Disclaimer: The convex optimization view described hereby is a cumulative result of the contribution of many research papers in online learning, to name just a few [Zin03, HKKA06, SSS07, AHR08, HK08]. The survey is intended as an advanced text. Prediction theory is a wide topic, for an in-depth introduction and wide treatment the reader is referred to [CBL06].

1.1 Outline

In the rest of this section we formally describe the online convex optimization model and the main algorithmic techniques as well as give a few simple example settings. In the next section we describe the RFTL family of algorithms and give a short analysis for regret bounds. In the third section we describe the primal-dual algorithm, show its equivalence to RFTL in certain settings, as well as give a modern analysis for its regret bounds.

1.2 The online convex optimization model

In online convex optimization, an online player iteratively chooses a point from a set in Euclidean space denoted $\mathcal{K} \subseteq \mathbb{R}^n$. Following Zinkevich [Zin03], we assume that the set $\mathcal{K}$ is non-empty, bounded and closed. For algorithmic-efficiency reasons that will be apparent later, we also assume the set $\mathcal{K}$ to be convex.

We denote the number of iterations by T (which is unknown to the online player). At iteration t, the online player chooses $\mathbf{x}_t \in \mathcal{K}$. After committing to this choice, a convex cost function $f_t : \mathcal{K} \mapsto \mathbb{R}$ is revealed. The cost incurred to the online player is the value of the cost function at the point she committed to $f_t(\mathbf{x}_t)$. Henceforth we consider mostly *linear* cost functions, and abuse notation to write $\mathbf{f}_t(x) = \mathbf{f}_t^\top x$.

The regret of the online player using algorithm $\mathcal{A}$ at time T, is defined to be the total cost minus the cost of the best single decision, where the best is chosen with the benefit of hindsight. We are usually interested in an upper bound on the worst case guaranteed regret, denoted

$$\text{Regret}_T(\mathcal{A}) = \sup_{\{f_1,\ldots,f_t\}} \left\{ \mathbf{E}[\textstyle\sum_{t=1}^T f_t(\mathbf{x}_t)] - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^T f_t(\mathbf{x}) \right\}.$$

Regret is the de-facto standard in measuring performance of learning algorithms. [1]

Intuitively, an algorithm attains non-trivial performance if its regret is sublinear as a function of T, i.e. $\text{Regret}_T(\mathcal{A}) = o(T)$, since this implies that "on the average" the algorithm performs as good as the best fixed strategy in hindsight.

[1]For some problems it is more natural to talk of "payoff" given to the online player rather than cost she incurs. If so, the payoff functions need to be concave and regret is defined analoguously, details are left to the reader.

The running time of an algorithm for online game playing is defined to be the worst-case expected time to produce $\mathbf{x}_t$, for an iteration $t \in [T]$ [2] in a T iteration repeated game. Typically, the running time will depend on n, T and parameters of the cost functions and underlying convex set.

1.3 Examples

1.3.1 The experts problem

Perhaps the most well known problem in prediction theory is the so called "experts problem". The decision maker has to choose from the advice of n given experts. After choosing one, a loss between zero and one is occurred. This scenario is repeated iteratively, each iteration the costs of the various experts are arbitrary. The goal is to do as the best expert in hindsight.

The online convex optimization problem captures this problem easily: the set of decision is the set of all distributions over n items, i.e. the n-dimensional simplex $\mathcal{K} = \Delta_n = \{x \in \mathbb{R}^n, \sum_i x_i = 1 \, , x_i \geq 0\}$. Let the cost to the i'th expert at iteration t be denoted by $\mathbf{f}_t(i)$. Then the cost functions are given by $\mathbf{f}_t(x) = \mathbf{f}_t^\top \mathbf{x}$ - this is the expected cost of playing according to distribution x.

1.3.2 Online shortest paths

In the online shortest path problem the decision maker is given a directed graph $G = (V, E)$ and a source-sink pair $s, t \in V$. At each iteration $t = 1$ to T, the decision maker chooses a path $p_t \in \mathcal{P}_{s,t}$, where $\mathcal{P}_{s,t} \subseteq \{E\}^{|V|}$ is the set of all s,t-paths in the graph. The adversary independently chooses weights on the edges of the graph $\mathbf{f}_t \in \mathbb{R}^m$. The decision maker suffers and observes loss, which is the weighted length of the chosen path $\sum_{e \in p_t} \mathbf{f}_t(e)$.

The discrete description of this problem as an experts problem, where we have an expert for every path, is inherently inefficient [3]. Instead, to model the problem as an online convex optimization problem, recall the standard description of the set of all distributions over paths (flows) in graph as a convex set in $\mathbb{R}^m$, with $O(m + |V|)$ constraints. Denote this flow polytope by $\mathcal{K}$. The expected cost of a given flow $\mathbf{x} \in \mathcal{K}$ (distribution over paths) is then a linear function, given by $\mathbf{f}_t^\top \mathbf{x}$, where $\mathbf{f}_t(e)$ is the length of the edge $e \in E$.

1.4 Algorithms for OCO

Algorithms for online convex optimization can be derived from the rich algorithmic techniques developed for prediction in various statistical and machine learning settings. The main purpose of this survey is, however, to describe two general algorithmic frameworks from which most, if not all, previous algorithms can be derived as special cases.

Perhaps the most straightforward approach to apply is for the online player to use whatever decision (point in the convex set) that would have been optimal till now. Formally, let

$$\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{K}} \sum_{i=1}^{t-1} \mathbf{f}_i(\mathbf{x})$$

The name given by Kalai and Vempala [KV05] to this Naive strategy is "Follow the Leader" (FTL), for obvious reasons. As Kalai and Vempala point out, this strategy fails miserably in a worst-case sense. That is, it's regret can be linear in the number of iterations, as the following example shows: Consider K to be

[2] here and henceforth we denote by $[n]$ the set of integers $\{1, ..., n\}$

[3] disclaimer: there are efficient algorithms which overcome this difficulty and do take this approach, at least for some settings. Yet in this survey we prefer the approach, which we find more natural, of modeling the structural problem as a polytope in low dimensional space.

the real line segment between minus one and one, and $\mathbf{f}_0 = \frac{1}{2}\mathbf{x}$, and let $\mathbf{f}_i$ be alternatively $-\mathbf{x}$ or $\mathbf{x}$. The FTL strategy will keep shifting between minus one and one, always making the wrong choice.

Kalai and Vempala proceed to analyze a modification of FTL with added noise to "stabilize" the decision (this modification is originally due to Hannan [Han57]). Similarly, much more general and varied twists on this basic FTL strategy can be conjured, and as we shall show also analyzed successfully. This is the essence of the RFTL meta-algorithm defined in this section.

Another natural approach one might take to online convex optimization is an iterative approach: start with some decision $\mathbf{x} \in \mathcal{K}$, and iteratively modify it according to the cost functions that are encountered. Some natural update rules include the gradient update, updates based on a multiplicative rule, on Newton's method, and so forth. Indeed, all of these suggestions make for useful algorithms. But as we shall show, they can all be seen as special cases of the same RFTL methodology !

2 The RFTL algorithm and its analysis

The motivation for taking the FTL approach was described above, and so was the caveat with straightforward implementation: as in the bad example we have considered, the prediction of FTL may vary wildly from one iteration to the next. Since our goal is regret, or comparison with a *fixed* point in the set, one would expect the algorithm to some form of convergence.

Indeed, this is the motivation for modifying the basic FTL strategy, and adding a *regularization* term, hence RFTL (Regularized Follow the Leader). The Regularizing function's purpose is exactly to add some stability to the prediction, and as such most any curved function with optimum inside of the feasible set shall do.

We proceed to formally describe the RFTL algorithmic template, and analyze it. The following we assume linear cost functions, $\mathbf{f}(\mathbf{x}) = \mathbf{f}^T\mathbf{x}$. If the cost functions are convex, one can use the inequality $\mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}^*) \leq \nabla\mathbf{f}_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{x}^*)$, and rename the function $\mathbf{f}_t(\mathbf{x})$ to $\mathbf{f}_t(\mathbf{x}) = \nabla\mathbf{f}_t(\mathbf{x}_t)^\top\mathbf{x}$, which is now linear.

2.1 Algorithm definition

Algorithm 1 RFTL

1: Input: $\eta > 0$, strongly convex regularizer function $\mathcal{R}$, and a convex compact set $\mathcal{K}$.
2: Let $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} [\mathcal{R}(\mathbf{x})]$.
3: **for** $t = 1$ to T **do**
4: Predict $\mathbf{x}_t$.
5: Observe the payoff function $\mathbf{f}_t$.
6: Update

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \underbrace{\left[\eta \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x})\right]}_{\Phi_t(\mathbf{x})} \tag{1}$$

7: **end for**

2.2 Special cases: multiplicative updates and gradient descent

Two famous algorithms which are captured by the above algorithm are so called the ”multiplicative update” algorithm and the gradient descent method. If $\mathcal{K} = \Delta_n$, then taking $\mathcal{R}(\mathbf{x}) = -\mathbf{x}\log\mathbf{x}$ gives a multiplicative update algorithm, in which

$$\mathbf{x}_{t+1} = \frac{\mathbf{x}_t \cdot \mathrm{diag}(e^{\eta \mathbf{f}_t})}{\|\mathbf{x}_t \cdot \mathrm{diag}(e^{\eta \mathbf{f}_t})\|_1}$$

If $\mathcal{K}$ is the unit ball and $\mathcal{R}(\mathbf{x}) = \|\mathbf{x}\|_2^2$, we get the gradient descent algorithm, in which

$$\mathbf{x}_{t+1} = \frac{\mathbf{x}_t - \eta \mathbf{f}_t}{\|\mathbf{x}_t - \eta \mathbf{f}_t\|_2}$$

This is easily seen in the next section where we give an equivalent definition of RFTL.

2.3 The regret bound

Henceforth we make use of general matrix norms. A PSD matrix $A \succeq 0$ gives rise to the norm $\|x\|_A = \sqrt{x^T A x}$. The *dual* norm of this matrix norm is $\|x\|_{A^{-1}} = \|x\|_A^*$. The generalized Cauchy-Schwartz theorem asserts $x \cdot y \leq \|x\|_A \|y\|_A^*$. We usually take A to be the Hessian of the regularization function $\mathcal{R}(x)$, denoted $\nabla^2 \mathcal{R}(x)$. In this case, we shorthand the notation to be $\|x\|_{\nabla^2 \mathcal{R}(y)} = \|x\|_y$, and similarly $\|x\|_{\nabla^{-2}\mathcal{R}(y)} = \|x\|_y^*$. Denote

$$\lambda_t = \max_{\mathbf{f}_t\ ,\ \mathbf{x} \in \mathcal{K}} \mathbf{f}_t^{\top} [\nabla^2 \mathcal{R}(\mathbf{x})]^{-1} \mathbf{f}_t \ \ , \ \ D = \max_{\mathbf{u} \in \mathcal{K}} \mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)$$

and

$$\lambda = \max_t \lambda_t \ .$$

Theorem 2.1. *The algorithm above achieves for every* $\mathbf{u} \in \mathcal{K}$ *the following bound on the regret:*

$$\mathit{Regret}_T = \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t - \mathbf{u}) \leq 2\sqrt{2\lambda D T} \ .$$

And in more detailed form:

Theorem 2.2. *The algorithm above achieves for every* $\mathbf{u} \in \mathcal{K}$ *the following bound on the regret:*

$$\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t - \mathbf{u}) \leq \eta \cdot 2 \sum_t \|\mathbf{f}_t\|_{\mathbf{z}_t}^{*\,2} + \frac{1}{\eta}\left[\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)\right]$$

To prove this theorem, we first relate the regret to the ”stability” in prediction. This is formally captured by the FTL-BTL lemma below, which holds in the aforementioned general scenario.

Lemma 2.3 (FTL-BTL Lemma)**.** *For every* $\mathbf{u} \in \mathcal{K}$*, the algorithm defined by* (1) *enjoys the following regret guarantee*

$$\sum_{t=1}^{T} \mathbf{f}_t^{\top}(\mathbf{x}_t - \mathbf{u}) \quad \leq \quad \sum_{t=1}^{T} \mathbf{f}_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{1}{\eta}[\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)]$$

We defer the proof of this simple lemma to the appendix, and proceed with the (short) proof of the main theorem.

Main Theorem. Now recall that $\mathcal{R}(x)$ is a convex function and $\mathcal{K}$ is convex. Then by Taylor expansion at $\mathbf{x}_{t+1}$, there exists a $\mathbf{z}_t \in [\mathbf{x}_{t+1}, \mathbf{x}_t]$ for which

$$\begin{aligned} \Phi_t(\mathbf{x}_t) &= \Phi_t(\mathbf{x}_{t+1}) + (\mathbf{x}_t - \mathbf{x}_{t+1})^\top \nabla \Phi_t(\mathbf{x}_{t+1}) + \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{z}_t}^2 \\ &\geq \Phi_t(\mathbf{x}_{t+1}) + \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{z}_t}^2 \end{aligned}$$

Recall our notation $\|\mathbf{y}\|_{\mathbf{z}}^2 = \mathbf{y}^\top \nabla^2 \Phi_t(\mathbf{z})\mathbf{y}$ and it follows that $\|\mathbf{y}\|_{\mathbf{z}}^2 = \mathbf{y}^\top \nabla^2 \mathcal{R}(\mathbf{z})\mathbf{y}$. The inequality above is true because $\mathbf{x}_{t+1}$ is a minimum of Φ_t over $\mathcal{K}$. Thus,

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{z}_t}^2 &\leq 2\,\Phi_t(\mathbf{x}_t) - 2\,\Phi_t(\mathbf{x}_{t+1}) \\ &= 2\,\left(\Phi_{t-1}(\mathbf{x}_t) - \Phi_{t-1}(\mathbf{x}_{t+1})\right) + 2\eta \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq 2\,\eta\, \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})\,. \end{aligned}$$

By the generalized Cauchy-Schwartz inequality,

$$\begin{aligned} \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) &\leq \|\mathbf{f}_t\|_{\mathbf{z}_t}^* \cdot \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{z}_t} \qquad \text{general CS} \qquad (2) \\ &\leq \|\mathbf{f}_t\|_{\mathbf{z}_t}^* \cdot \sqrt{2\,\eta\, \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})} \end{aligned}$$

Shifting sides and squaring we get

$$\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 2\,\eta\, \|\mathbf{f}_t\|_{\mathbf{z}_t}^{*\,2}$$

This together with the FTL-BTL Lemma, summing over T periods we obtain the Theorem. Choosing the optimal η, we obtain

$$R_T \leq \min_\eta \left\{ 2\,\eta\lambda T + \frac{1}{\eta}[\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)] \right\} \leq 2\sqrt{2\,D\,\lambda\,T}\,.$$

□

3 The "primal-dual" approach

The other approach for proving regret bounds, which we call "primal-dual", originates from the so called "link-function methodology", as introduced in [GLS01, KW01]. A central concept useful for this method are Bregman divergences, formally defined by,

Definition 3.1. Denote by $B^{\mathcal{R}}(\mathbf{x}||\mathbf{y})$ the Bregman divergence with respect to the function $\mathcal{R}$, defined as

$$B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}) = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - (\mathbf{x} - \mathbf{y})^\top \nabla \mathcal{R}(\mathbf{y})\,.$$

As opposed to the "one shot" nature of RFTL, the primal-dual algorithm is an iterative algorithm, which computes the next prediction using a simple update rule and the previous prediction. The generality of the method stems from the update being carried out in a "dual" space, where the duality notion is defined by the choice of regularization.

Algorithm 2 Primal-dual

Let $\mathcal{K}$ be a convex set

Input: parameter $\eta > 0$, regularizer function $\mathcal{R}(\mathbf{x})$.

for $t = 1$ to T **do**

  If $t = 1$, choose $\mathbf{y}_1$ such that $\nabla\mathcal{R}(\mathbf{y}_1) = \mathbf{0}$.

  If $t > 1$, choose $\mathbf{y}_t$ such that $\nabla\mathcal{R}(\mathbf{y}_t) = \nabla\mathcal{R}(\mathbf{y}_{t-1}) - \eta\, \mathbf{f}_{t-1}$.

  Project according to $B^{\mathcal{R}}$:

$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{K}} B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}_t)$$

end for

3.1 Equivalence to RFTL in the linear setting

For the special case of linear cost functions, the algorithm above and RFTL are identical, as we show now. The primal-dual algorithm, however, can be analyzed in a very different way, which is extremely useful in certain online scenarios.

Lemma 3.2. *For linear cost functions, the primal-dual and RFTL algorithms produce identical predictions, i.e.,*

$$\arg\min_{x\in\mathcal{K}} \left(\mathbf{f}_t^{\mathsf{T}}\mathbf{x} + \frac{1}{\eta}\mathcal{R}(\mathbf{x}) \right) = \arg\min_{\mathbf{x}\in\mathcal{K}} B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}_t)\ .$$

Proof. First, observe that the unconstrained minimum

$$\mathbf{x}_t^* \equiv \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \sum_{s=1}^{t-1} \mathbf{f}_s^{\mathsf{T}}\mathbf{x} + \frac{1}{\eta}\mathcal{R}(\mathbf{x}) \right\}$$

satisfies

$$\sum_{s=1}^{t-1} \mathbf{f}_s + \frac{1}{\eta}\nabla\mathcal{R}(\mathbf{x}_t^*) = \mathbf{0}\ .$$

Since $\mathcal{R}(\mathbf{x})$ is strictly convex, there is only one solution for the above equation and thus $\mathbf{y}_t = \mathbf{x}^*$. Hence,

$$\begin{aligned} B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}_t) &= \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}_t) - (\nabla\mathcal{R}(\mathbf{y}_t))^{\mathsf{T}}(\mathbf{x} - \mathbf{y}_t) \\ &= \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}_t) + \eta \sum_{s=1}^{t-1} \mathbf{f}_s^{\mathsf{T}}(\mathbf{x} - \mathbf{y}_t)\ . \end{aligned}$$

Since $\mathcal{R}(\mathbf{y}_t)$ and $\sum_{s=1}^{t-1} \mathbf{f}_s^{\mathsf{T}}\mathbf{y}_t$ are independent of $\mathbf{x}$), it follows that $B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}_t)$ is minimized at the point $\mathbf{x}$ that minimizes $\mathcal{R}(\mathbf{x}) + \eta \sum_{s=1}^{t-1} \mathbf{f}_s^{\mathsf{T}}\mathbf{x}$ over $\mathcal{K}$ which, in turn, implies that

$$\arg\min_{\mathbf{x}\in\mathcal{K}} B^{\mathcal{R}}(\mathbf{x}||\mathbf{y}_t) = \arg\min_{\mathbf{x}\in K} \left\{ \sum_{s=1}^{t-1} \mathbf{f}_s^{\mathsf{T}}\mathbf{x} + \frac{1}{\eta}\mathcal{R}(\mathbf{x}) \right\}\ .$$

□

3.2 Regret bounds for the primal-dual algorithm

Theorem 3.3. *Suppose that h is such that $B_{\mathcal{R}}(\mathbf{x},\mathbf{y}) \geq \|\mathbf{x}-\mathbf{y}\|^2$ for some norm $\|\cdot\|$. Let $\|\nabla \mathbf{f}_t(\mathbf{x}_t)\|^* \leq G^*$ for all t, and $\forall \mathbf{x} \in K$.$B_{\mathcal{R}}(\mathbf{x},\mathbf{x}_1) \leq D^2$. Applying the primal-dual algorithm with $\eta = \frac{D}{G_*\sqrt{T}}$, we have*

$$\textit{Regret} \leq DG^*\sqrt{T}$$

Proof. The proof follows that of ogd, with the Bregman divergence replacing the Euclidean distance as a potential function. As the functions $\mathbf{f}_t$ are convex, for any $\mathbf{x}^* \in K$,

$$\mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}^*) \leq \nabla \mathbf{f}_t(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^*).$$

The following property of Bregman divergences follows easily from the definition: for any vectors x, y, z,

$$(\mathbf{x}-\mathbf{y})^\top (\nabla \mathcal{R}(\mathbf{z}) - \nabla \mathcal{R}(\mathbf{y})) = B_{\mathcal{R}}(\mathbf{x},\mathbf{y}) - B_{\mathcal{R}}(\mathbf{x},\mathbf{z}) + B_{\mathcal{R}}(\mathbf{y},\mathbf{z}).$$

Combining both observations,

$$\begin{aligned}
2(\mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}^*)) &\leq 2\nabla \mathbf{f}_t(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^*) \\
&= \frac{1}{\eta}(\nabla \mathcal{R}(\mathbf{y}_{t+1}) - \nabla \mathcal{R}(\mathbf{x}_t))^\top (\mathbf{x}^* - \mathbf{x}_t) \\
&= \frac{1}{\eta}[B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{x}_t) - B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{y}_{t+1}) + B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1})] \\
&\leq \frac{1}{\eta}[B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{x}_t) - B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{x}_{t+1}) + B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1})]
\end{aligned}$$

where the last inequality follows from the Pythagorean Theorem for Bregman divergences [CBL06], as x_{t+1} is the projection w.r.t the Bregman divergence of y_{t+1} and $\mathbf{x}^* \in K$ is in the convex set. Summing over all iterations,

$$\begin{aligned}
2\text{Regret} \quad &\leq \tfrac{1}{\eta}[B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{x}_1) - B_{\mathcal{R}}(\mathbf{x}^*,\mathbf{x}_T)] + \textstyle\sum_{t=1}^T \tfrac{1}{\eta} B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1}) \\
&\leq \tfrac{1}{\eta}D^2 + \textstyle\sum_{t=1}^T \tfrac{1}{\eta} B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1}) \qquad (3)
\end{aligned}$$

We proceed to bound $B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1})$. By definition of Bregman divergence, and the dual norm inequality stated before,

$$\begin{aligned}
B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1}) + B_{\mathcal{R}}(\mathbf{y}_{t+1},\mathbf{x}_t) &= (\nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{y}_{t+1}))^\top (\mathbf{x}_t - \mathbf{y}_{t+1}) \\
&= 2\eta \nabla \mathbf{f}_t(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{y}_{t+1}) \\
&\leq \eta^2 G^{*\,2} + \|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2.
\end{aligned}$$

Thus, by our assumption $B_{\mathcal{R}}(\mathbf{x},\mathbf{y}) \geq \|\mathbf{x}-\mathbf{y}\|^2$, we have

$$B_{\mathcal{R}}(\mathbf{x}_t,\mathbf{y}_{t+1}) \leq \eta^2 G^{*\,2} + \|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2 - B_{\mathcal{R}}(\mathbf{y}_{t+1},\mathbf{x}_t) \leq \eta^2 G^{*\,2}.$$

Plugging back into Equation (3), and by non-negativity of the Bregman divergence, we get

$$\text{Regret} \leq \frac{1}{2}[\frac{1}{\eta}D^2 + \eta T G_{*\,2}] \leq DG^*\sqrt{T}.$$

By taking $\eta = \frac{D}{2\sqrt{T}G^*}$

□

It is now clear to see that taking $\mathcal{R}$ to be the negation of the entropy function, $\mathcal{R}(\mathbf{x}) = \mathbf{x}\log\mathbf{x}$, we get the multiplicative update algorithm (over the simplex). It is well known that the entropy function satisfies that $B_R(\mathbf{x}, \mathbf{y}) \geq \|\mathbf{x} - \mathbf{y}\|_1$. This implies that the regret of the above algorithm with the entropy regularization satisfies over the simplex that

$$\text{Regret} \leq G_\infty \sqrt{T \log n}$$

Taking $\mathcal{R} = \frac{1}{2}\|\mathbf{x}\|_2^2$ we obtain the gradient descent algorithm.

References

[AHR08] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In Rocco A. Servedio and Tong Zhang, editors, *COLT*, pages 263–274. Omnipress, 2008.

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

[GLS01] A .J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

[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.

[HKKA06] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In Gábor Lugosi and Hans-Ulrich Simon, editors, *COLT*, volume 4005 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 2006.

[HK08] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In Rocco A. Servedio and Tong Zhang, editors, *COLT*, pages 57–68. Omnipress, 2008.

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

[KW01] J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[SZ08] Rocco A. Servedio and Tong Zhang, editors. *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*. Omnipress, 2008.

[SSS07] Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

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

A The FTL-BTL Lemma

Here is the promised proof of the FTL-BTL lemma:

proof of Lemma 2.3. For convenience, denote by $\mathbf{f}_0 = \frac{1}{\eta}\mathcal{R}$, and assume we start the algorithm from $t = 0$ with an arbitrary $\mathbf{x}_0$. The lemma is now proved by induction on T.
Induction base: Note that by definition, we have that $\mathbf{x}_1 = \arg\min_{\mathbf{x}}\{\mathcal{R}(\mathbf{x})\}$, and thus $\mathbf{f}_0(\mathbf{x}_1) \leq \mathbf{f}_0(\mathbf{u})$ for all $\mathbf{u}$, thus $\mathbf{f}_0(\mathbf{x}_0) - \mathbf{f}_0(\mathbf{u}) \leq \mathbf{f}_0(\mathbf{x}_0) - \mathbf{f}_0(\mathbf{x}_1)$.
Induction step: Assume that that for T, we have

$$\sum_{t=0}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{u}) \quad \leq \quad \sum_{t=0}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{t+1})$$

and let us prove for $T + 1$. Since $\mathbf{x}_{T+2} = \arg\min_{\mathbf{x}}\{\sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{x})\}$ we have:

$$\begin{aligned}
\sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{x}_t) - \sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{u}) &\leq \textstyle\sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{x}_t) - \sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{x}_{T+2}) \\
&= \textstyle\sum_{t=0}^{T}(\mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{T+2})) + \mathbf{f}_{T+1}(\mathbf{x}_{T+1}) - \mathbf{f}_{T+1}(\mathbf{x}_{T+2}) \\
&\leq \textstyle\sum_{t=0}^{T}(\mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{t+1})) + \mathbf{f}_{T+1}(\mathbf{x}_{T+1}) - \mathbf{f}_{T+1}(\mathbf{x}_{T+2}) \\
&= \textstyle\sum_{t=0}^{T+1} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{t+1})
\end{aligned}$$

Where in the third line we used the induction hypothesis for $\mathbf{u} = \mathbf{x}_{T+2}$. We conclude that

$$\begin{aligned}
\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{u}) &\leq \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{t+1}) + [-\mathbf{f}_0(\mathbf{x}_0) + \mathbf{f}_0(\mathbf{u}) + \mathbf{f}_0(\mathbf{x}_0) - \mathbf{f}_0(\mathbf{x}_1)] \\
&= \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}_{t+1}) + \frac{1}{\eta}\left[\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)\right]
\end{aligned}$$

□