The convex optimization approach to regret minimization

Elad Hazan
Technion - Israel Institute of Technology
ehazan@ie.technion.ac.il

Abstract

A well studied and general setting for prediction and decision making is regret minimization in games. Recently the design of algorithms in this setting has been influenced by tools from convex optimization. In this chapter we describe the recent framework of *online convex optimization* which naturally merges optimization and regret minimization. We describe the basic algorithms and tools at the heart of this framework, which have led to the resolution of fundamental questions of learning in games.

Contents

1 Introduction

In the online decision making scenario, a player has to choose from a pool of available decisions and then incurs a loss corresponding to the quality of decision made. The regret minimization paradigm suggests the goal of incurring an average loss which approaches that of the best fixed decision in hindsight. Recently tools from convex optimization have given rise to algorithms which are more general, unifying previous results, and many times giving new and improved regret bounds.

In this chapter we survey some of the recent developments in this exciting merger of optimization and learning. We start by describing two general templates for producing algorithms and proving regret bounds. The templates are very simple, and unify the analysis of many previous well-known and used algorithms (i.e. multiplicative weights and gradient descent). For the setting of online linear optimization, we also prove that the two templates are equivalent.

After describing the framework and algorithmic templates, we describe some successful applications: characterization of regret bounds in terms of convexity of loss functions, bandit linear optimization and variational regret bounds.

1.1 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 [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 $\mathbf{f}_t : \mathcal{K} \mapsto \mathbb{R}$ is revealed. The cost incurred to the online player is the value of the cost function at the point she committed to $\mathbf{f}_t(\mathbf{x}_t)$. Henceforth we consider mostly *linear* cost functions, and abuse notation to write $\mathbf{f}_t(\mathbf{x}) = \mathbf{f}_t^\top \mathbf{x}$.

The feedback available to the player falls into two main categories: in the full information model, all information about the function $\mathbf{f}_t$ is observable by the player (after incurring the loss). In the "bandit" model, the player only observes the loss $\mathbf{f}_t(\mathbf{x}_t)$ itself.

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 fixed 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

$$\text{Regret}_T(\mathcal{A}) = \sup_{\{\mathbf{f}_1,\ldots,\mathbf{f}_t\}} \left\{ \mathbf{E}[\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t)] - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}) \right\}.$$

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

Intuitively, an algorithm performs well 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 well as the best fixed strategy in hindsight.

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 the parameters of the cost functions and underlying convex set.

1.2 Examples

1.2.1 Prediction from experts advice

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 incurred. This scenario is repeated iteratively, and at each iteration the costs of the various experts are arbitrary. The goal is to do as well as the best expert in hindsight.

The online convex optimization problem captures this problem as a special case: the set of decisions is the set of all distributions over n elements (experts), i.e. the n-dimensional simplex $\mathcal{K} = \Delta_n = \{\mathbf{x} \in \mathbb{R}^n, \sum_i \mathbf{x}_i = 1\ , \mathbf{x}_i \geq 0\}$. 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 choosing an expert according to distribution $\mathbf{x}$, and happens to be linear.

1.2.2 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 $t \in [T]$, the decision maker chooses a path $p_t \in \mathbb{P}_{s,t}$, where $\mathbb{P}_{s,t} \subseteq \{E\}^{|V|}$ is the set of all $s-t$-paths in the graph. The adversary independently chooses weights on the edges of the graph, given by a function from the edges to the reals $\mathbf{f}_t : E \mapsto \mathbb{R}$, which can be represented as a vector in m-dimensional space (for $m = |E|$): $\mathbf{f}_t \in \mathbb{R}^m$.

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

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

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, presents an efficiency challenge: there are potentially exponentially many paths in terms of the graph representation size. Much work has been devoted to resolve this efficiency issue, and efficient algorithms have been found in this discrete formulation, e.g. [TW03, KV05, AK08]. However, the optimal regret bound for the bandit version of this problem eluded researchers for some time, and was finally resolved only within the online convex optimization framework [AHR08, DHK08].

The online shortest path problem can be cast in the online convex optimization framework as follows. 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 $e \in E$. This succinct formulation inherently leads to computationally efficient algorithms.

1.2.3 Portfolio selection

The universal portfolio selection problem which we briefly describe henceforth is due to [Cov91]. At each iteration $t = 1$ to T, the decision maker chooses a distribution of her wealth over n assets $\mathbf{x}_t \in \Delta_n$. The adversary independently chooses market returns for the assets, i.e. a vector $\mathbf{r}_t \in \mathbb{R}_+^n$ such that each coordinate $\mathbf{r}_t(i)$ is the price ratio for the i'th asset between the iterations t and $t + 1$. The ratio between the wealth of the investor at iterations $t + 1$ and t is $\mathbf{r}_t^\top \mathbf{x}_t$, and hence the gain in this setting is defined to be the logarithm of this change ratio in wealth $\log(\mathbf{r}_t^\top \mathbf{x}_t)$. Notice that since $\mathbf{x}_t$ is the distribution of the investor's wealth, even if $\mathbf{x}_{t+1} = \mathbf{x}_t$, the investor may still need to trade to adjust for price changes.

The goal of regret minimization, which in this case corresponds to minimizing the difference $\max_{\mathbf{x} \in \Delta_n} \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}) - \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}_t)$, has an intuitive interpretation. The first term is the logarithm of the wealth accumulated by the distribution $\mathbf{x}^*$. Since this distribution is fixed, it corresponds to a strategy of rebalancing the position after every trading period, and hence called a *constant rebalanced portfolio*. The second expression is the logarithm of the wealth accumulated by the online decision maker. Hence regret minimization corresponds to maximizing the ratio of investor wealth vs. wealth of the best benchmark from a pool of investing strategies.

A *universal* portfolio selection algorithm is defined to be one that attains regret converging to zero in this setting. Such an algorithm, albeit requiring exponen-

tial time, was first described in [Cov91]. The online convex optimization framework has given rise to much more efficient algorithms based on Newton's method [HAK07].

1.3 Algorithms for online convex optimization

Algorithms for online convex optimization can be derived from rich algorithmic techniques developed for prediction in various statistical and machine learning settings. Henceforth we describe two general algorithmic frameworks from which many previous algorithms can be derived as special cases.

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

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

This flavor of strategy is known as "fictitious play" in economics, and was named "Follow the Leader" (FTL) by [KV05]. As the latter paper points 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 the real line segment between minus one and one, and $\mathbf{f}_1 = \frac{1}{2}\mathbf{x}$, and let $\mathbf{f}_i$ alternate between $-\mathbf{x}$ or $\mathbf{x}$. The FTL strategy will keep shifting between -1 and $+1$, 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 [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 meta-algorithm defined in this section.

Another natural approach for 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 general methodology we analyze next!

2 The RFTL algorithm and its analysis

Recall the caveat with straightforward use of follow-the-leader: as in the bad example we have considered, the prediction of FTL may vary wildly from one iteration to the next. This motivates the modification of the basic FTL strategy in order

to stabilize the prediction. By adding a *regularization* term, we obtain the RFTL (Regularized Follow the Leader) algorithm.

We proceed to formally describe the RFTL algorithmic template, and analyze it. While the analysis given is optimal asymptotically, we do not give the best constants possible in order to simplify presentation.

In this section we consider only linear cost functions, $\mathbf{f}(\mathbf{x}) = \mathbf{f}^T\mathbf{x}$. The case of convex cost functions can be reduced to the linear case via 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 considering the function $\hat{\mathbf{f}}_t(\mathbf{x}) = \nabla\mathbf{f}_t(\mathbf{x}_t)^\top\mathbf{x}$, which is now linear.

2.1 Algorithm definition

The generic RFTL meta-algorithm is defined in figure 1 below. The regularization function $\mathcal{R}$ is assumed to be strongly convex and smooth such that it has a continuous second derivative.

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 $\mathcal{K} = \Delta_n = \{\mathbf{x} \geq 0\ ,\ \sum_i \mathbf{x}(i) = 1\}$, then taking $\mathcal{R}(\mathbf{x}) = \mathbf{x}\log\mathbf{x}$ gives a multiplicative update algorithm, in which

$$\mathbf{x}_{t+1}(i) = \frac{\mathbf{x}_t(i) \cdot e^{\eta\mathbf{f}_t(i)}}{\sum_{i=1}^{n} \mathbf{x}_t(i) \cdot e^{\eta\mathbf{f}_t(i)}}$$

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

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

It is possible to derive these special cases by the KKT optimality conditions of Equation 1. However, we give an easier proof of these facts in the next section, in which we give an equivalent definition of RFTL for the case of linear cost functions.

2.3 The regret bound

Henceforth we make use of general matrix norms. A PSD matrix $A \succ 0$ 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

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

Notice that both λ and D depend on the regularization function, the convex decision set, and the magnitude of the cost functions. Intuitively, the term D corresponds to the diameter of the set $\mathcal{K}$ as measured by the regularization $\mathcal{R}$, while the term λ corresponds to the (squared) magnitude of the cost functions, measured according to a norm which is derived from the regularization.

Theorem 1. *The algorithm above achieves for every $\mathbf{u} \in \mathcal{K}$ the following bound on the regret:*

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

Consider the expert problem for example: the convex set is the simplex, take $\mathcal{R}$ to be the negative entropy function (which corresponds to the multiplicative update algorithm), and the costs are bounded by one in each coordinate. Then $\mathbf{f}^{\mathsf{T}} [\nabla^2 \mathcal{R}(\mathbf{x})]^{-1} \mathbf{f} = \sum_i \mathbf{f}(i)^2 \mathbf{x}(i) \leq \sum_i \mathbf{x}(i) = 1$ which implies $\lambda \leq 1$. The parameter D in this case is bounded by $\max_{\mathbf{u} \in \Delta} \sum_i \mathbf{u}(i) \log \frac{1}{\mathbf{u}(i)} \leq \log n$. This gives a regret bound of $O(\sqrt{T \log n})$, which is known to be tight. [3]

[3] In the case of multiplicative updates, as well as in other regularization functions of interest, it is possible to obtain a tighter bound in Theorem 1: the term λ can be redefined as $\lambda = \max_t \mathbf{f}_t^{\mathsf{T}} [\nabla^2 \mathcal{R}(\mathbf{x}_t)]^{-1} \mathbf{f}_t$. The derivation is not in the scope of this survey, see [AHR08] for more details.

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

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

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

Proof of Main Theorem. Recall that $\mathcal{R}(x)$ is a convex function and $\mathcal{K}$ is convex. Then by Taylor expansion (with its explicit remainder term via the mean-value theorem) at $\mathbf{x}_{t+1}$, there exists a $\mathbf{z}_t \in [\mathbf{x}_{t+1}, \mathbf{x}_t]$ for which

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

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

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

By the generalized Cauchy-Schwartz inequality,

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

Shifting sides and squaring we get

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

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], and related to the "mirrored descent" paradigm in the optimization community. A central concept useful for this method are Bregman divergences, formally defined by,

Definition 1. *Denote by $B_{\mathcal{R}}(\mathbf{x}||\mathbf{y})$ the Bregman divergence with respect to the function $\mathcal{R}$, defined as*

$$B_{\mathcal{R}}(\mathbf{x}||\mathbf{y}) = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - (\mathbf{x}-\mathbf{y})^{\mathsf{T}}\nabla\mathcal{R}(\mathbf{y}) \ .$$

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

1: Let $\mathcal{K}$ be a convex set
2: Input: parameter $\eta > 0$, regularizer function $\mathcal{R}(\mathbf{x})$.
3: **for** $t = 1$ to T **do**
4: If $t = 1$, choose $\mathbf{y}_1$ such that $\nabla\mathcal{R}(\mathbf{y}_1) = \mathbf{0}$.
5: If $t > 1$, choose $\mathbf{y}_t$ such that:

Lazy version: $\nabla\mathcal{R}(\mathbf{y}_t) = \nabla\mathcal{R}(\mathbf{y}_{t-1}) - \eta\,\mathbf{f}_{t-1}$.

Active version: $\nabla\mathcal{R}(\mathbf{y}_t) = \nabla\mathcal{R}(\mathbf{x}_{t-1}) - \eta\,\mathbf{f}_{t-1}$.

6: Project according to $B_{\mathcal{R}}$:

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

7: **end for**

3.1 Equivalence to RFTL in the linear setting

For the special case of linear cost functions, the algorithm above (lazy version) 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 2. *For linear cost functions, the lazy 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 $\mathbf{y}_t=\mathbf{x}_t^*$. 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 2. *Suppose that $\mathcal{R}$ is such that $B_{\mathcal{R}}(\mathbf{x},\mathbf{y})\geq\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$ 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 (active version) with $\eta=\frac{D}{2G_*\sqrt{T}}$, we have*

$$\textit{Regret}_T\leq DG_*\sqrt{T}$$

Proof. Since 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 $\mathbf{x}, \mathbf{y}, \mathbf{z}$,

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

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 generalized Pythagorean inequality (see [CBL06] Lemma 11.3), as $\mathbf{x}_{t+1}$ is the projection w.r.t the Bregman divergence of $\mathbf{y}_{t+1}$ and $\mathbf{x}^* \in K$ is in the convex set. Summing over all iterations,

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

We proceed to bound $B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1})$. By definition of Bregman divergence, and the generalized Cauchy-Schwartz inequality,

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

where in the last inequality follows from $(a - b)^2 \geq 0$. Thus, by our assumption $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2$, we have

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

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

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

by taking $\eta = \frac{D}{2\sqrt{T}G_*}$

□

3.3 Deriving the multiplicative update and gradient descent algorithms

We have stated in the previous section that by taking $\mathcal{R}$ to be the negative entropy function over the simplex, the RFTL template specializes to become a multiplicative updates algorithm. Since we have proved that RFTL is equivalent to the primal-dual algorithm, the same is true for the latter, and the same regret bound applies.

If $\mathcal{R}(\mathbf{x}) = \mathbf{x}\log\mathbf{x}$ is the negative entropy function, then $\nabla\mathcal{R}(\mathbf{x}) = \mathbf{1} + \log\mathbf{x}$, and hence the update rule for the primal-dual algorithm 2 (the lazy and adaptive versions are identical in this case) becomes:

$$\log\mathbf{y}_t = \log\mathbf{x}_{t-1} - \eta\mathbf{f}_{t-1}$$

or, $\mathbf{y}_t(i) = \mathbf{x}_{t-1}(i)\cdot e^{-\eta\mathbf{f}_{t-1}(i)}$. Since the entropy projection corresponds to scaling by the ℓ_1 norm, it follows that $\mathbf{x}_{t+1}(i) = \frac{\mathbf{x}_t(i)\cdot e^{\eta\mathbf{f}_t(i)}}{\sum_{i=1}^n \mathbf{x}_t(i)\cdot e^{\eta\mathbf{f}_t(i)}}$. The regret of the multiplicative updates algorithm can be bounded as in section 2.3 by $O(\sqrt{T\log n})$.

To derive the online gradient descent algorithm, take $\mathcal{R} = \frac{1}{2}\|\mathbf{x}\|_2^2$. In this case, $\nabla\mathcal{R}(\mathbf{x}) = \mathbf{x}$, and hence the update rule for the primal-dual algorithm 2 becomes:

$$\mathbf{y}_t = \mathbf{y}_{t-1} - \eta\mathbf{f}_{t-1}$$

and hence when $\mathcal{K}$ is the unit ball $\mathbf{x}_{t+1} = \frac{\mathbf{x}_1 - \eta\sum_{\tau=2}^t \mathbf{f}_\tau}{\|\mathbf{x}_1 - \eta\sum_{\tau=2}^t \mathbf{f}_\tau\|_2} = \frac{\mathbf{x}_t - \eta\mathbf{f}_t}{\|\mathbf{x}_t - \eta\mathbf{f}_t\|_2}$.

4 Convexity of loss functions

In this section we review one of the first consequences of the convex optimization approach to decision making. Namely, the characterization of attainable regret bounds in terms of convexity of loss functions. It has long been known that special kinds of loss functions permit tighter regret bounds than other loss functions. For example, in the portfolio selection problem Cover's algorithm attained regret which depends on the number of iterations T as $O(\log T)$. This is in contrast to online linear optimization, or the experts problem, in which $\Theta(\sqrt{T})$ is known to be tight.

In this section we give a simple gradient-descent based algorithm which attains logarithmic regret if the loss functions are *strongly convex*. Interestingly, the naive fictitious play (FTL) algorithm attains essentially the same regret bounds in this special case. Similar bounds are attainable under weaker conditions on the loss functions, which capture the portfolio selection problem, and have led to the aforementioned efficient algorithm for Cover's problem [HAK07].

We say that a function is α-strongly convex if its second derivative is strictly bounded away from zero. In higher dimensions this corresponds to the matrix inequality $\nabla^2 \mathbf{f}(\mathbf{x}) \succeq \alpha \cdot \mathbb{I}$, where $\nabla^2 \mathbf{f}(\mathbf{x})$ is the hessian of the function and $A \succeq B$ denotes that the matrix $A - B$ is positive semi-definite. For example, the squared loss, i.e. $\mathbf{f}(\mathbf{x}) = \|\mathbf{x} - \mathbf{a}\|_2^2$, is 1-strongly convex.

Algorithm 3 Online gradient descent

1: Input: convex set $\mathcal{K}$, initial point $\mathbf{x}_0 \in \mathcal{K}$, learning rates $\eta_1, ..., \eta_t$.
2: **for** $t = 1$ to T **do**
3: Let $\mathbf{y}_t = \mathbf{x}_{t-1} - \eta_{t-1} \nabla \mathbf{f}_{t-1}(\mathbf{x}_{t-1})$.
4: Project onto $\mathcal{K}$:

$$\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}_t\|_2$$

5: **end for**

The following theorem, proved in [HAK07], establishes logarithmic bounds on the regret if the cost functions are strongly convex. Denote by G an upper bound on the Euclidean norm of the gradients.

Theorem 3. *The online gradient descent algorithm with step sizes $\eta_t = \frac{1}{\alpha t}$ achieves the following guarantee, for all $T \geq 1$.*

$$\textit{Regret}_T(OGD) \leq \frac{G^2}{2\alpha}(1 + \log T)$$

Proof. Let $\mathbf{x}^* \in \arg\min_{\mathbf{x} \in \mathbb{P}} \sum_{t=1}^{T} f_t(\mathbf{x})$. Recall the definition of regret:

$$\text{Regret}_T(OGD) = \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}^*)$$

Denote $\nabla_t \triangleq \nabla \mathbf{f}_t(\mathbf{x}_t)$. By α-strong convexity, we have,

$$\begin{aligned} f_t(\mathbf{x}^*) &\geq f_t(\mathbf{x}_t) + \nabla_t^\top (\mathbf{x}^* - \mathbf{x}_t) + \frac{\alpha}{2}\|\mathbf{x}^* - \mathbf{x}_t\|^2 \\ 2(f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*)) &\leq 2\nabla_t^\top (\mathbf{x}_t - \mathbf{x}^*) - \alpha\|\mathbf{x}^* - \mathbf{x}_t\|^2 \end{aligned} \tag{4}$$

Following Zinkevich's analysis, we upper-bound $\nabla_t^\top(\mathbf{x}_t - \mathbf{x}^*)$. Using the update rule for $\mathbf{x}_{t+1}$ and the generalized Pythagorian inequality ([CBL06] Lemma 11.3), we get

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \;=\; \|\Pi(\mathbf{x}_t - \eta_t\nabla_t) - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_t - \eta_t\nabla_t - \mathbf{x}^*\|^2.$$

Hence,

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \quad\leq\quad \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \eta_t^2\|\nabla_t\|^2 - 2\eta_t\nabla_t^\top(\mathbf{x}_t - \mathbf{x}^*)$$

Shifting sides,

$$2\nabla_t^\top(\mathbf{x}_t - \mathbf{x}^*) \quad\leq\quad \frac{\|\mathbf{x}_t - \mathbf{x}^*\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2}{\eta_t} + \eta_t G^2 \tag{5}$$

Sum up (5) from $t = 1$ to T. Set $\eta_t = 1/(\alpha t)$, and using (4), we have:

$$\begin{aligned} 2\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \mathbf{f}_t(\mathbf{x}^*) \quad &\leq \quad \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \alpha\right) + G^2\sum_{t=1}^{T}\eta_t \\ &= \quad G^2\sum_{t=1}^{T}\frac{1}{\alpha t} \;\leq\; \frac{G^2}{\alpha}(1 + \log T) \end{aligned}$$

□

5 Recent Applications

In this section we describe to recent applications of the convex optimization view to regret minimization which have resolved open questions in the field.

5.1 Bandit linear optimization

The first application is to the bandit linear optimization problem: online linear optimization is a special case of online convex optimization in which the loss functions are linear (such as analyzed for the RFTL algorithm). In the bandit version, called bandit linear optimization, the only feedback available to the decision maker is the loss (rather than the entire loss function), and the knowledge that some unknown linear function generated this loss. This general framework naturally captures important problems such as online routing and online ad-placement for search engine results.

This generalization was put forth by [AK08] in the context of the online shortest path problem described previously. [AK08] gave an efficient algorithm for the

problem with a suboptimal regret bound, and conjectured the existence of an efficient and optimal-regret algorithm.

The problem attracted much attention in the machine learning community [FKM05, DH06, DHK07, BDH$^+$], until this question was finally resolved in [AHR08] where an efficient and optimal expected regret algorithm was described. Later [AR09] gave an efficient algorithm which also attains this optimal regret bound with high probability. The paper introduced the use of self-concordant barrier functions as a regularization in the RFTL framework. Self-concordant barriers are a powerful tool from optimization which has enabled researchers in operations research to develop efficient polynomial-time algorithms for (offline) convex optimization. The scope of this deep technical issue is beyond this survey, but the resolution of this open question is an excellent example of how the convex optimization approach to regret minimization led to the discovery of powerful tools which in turn resolved fundamental questions in machine learning.

5.2 Variational regret bounds

A cornerstone of modern machine learning are algorithms for prediction from expert advice, the first example of regret minimization we have described. It is already well established that there exists algorithms that, under fully adversarial cost sequences, attain average cost approaching that of the best expert in hindsight. More precisely, there exist efficient algorithms which attain regret of $O(\sqrt{T \log n})$ in the setting of prediction from expert advice with n experts.

However, *a priori* it is not clear why online learning algorithms should have high regret (growing with the number of iterations) in an unchanging environment. As an extreme example, consider a setting in which there are only two experts. Suppose that the first expert always incurs cost 1, whereas the second expert always incurs cost $\frac{1}{2}$. One would expect to "figure out" this pattern quickly, and focus on the second expert, thus incurring a total cost that is at most $\frac{T}{2}$ plus at most a constant extra cost (irrespective of the number of rounds T), thus having only constant regret. However, for a long time all analyses of expert learning algorithms only gave a regret bound of $\Theta(\sqrt{T})$ in this simple case (or very simple variations of it).

More generally, the natural bound on the regret of a "good" learning algorithm should depend on *variation* in the sequence of costs, rather than purely on the number of iterations. If the cost sequence has low variation, we expect our algorithm to be able to perform better.

This intuition has a direct analog in the stochastic setting: here, the sequence of experts' costs are independently sampled from a distribution. In this situation, a natural bound on the rate of convergence to the optimal expert is controlled by the

variance of the distribution (low variance should imply faster convergence). This conjecture was formalized by Cesa-Bianchi, Mansour and Stoltz (henceforth the "CMS conjecture") in [CBMS07].

The CMS conjecture was proved in the more general case of online linear optimization in [HK10]. Again, the convex optimization view was instrumental in the solution, and taken the general linear optimization view it was found that a simple geometric argument implies the result. Further work on variational bounds included an extension to the bandit linear optimization setting [HK09a] and to exp-concave loss functions including the problem of portfolio selection [HK09b].

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.

[AK08] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *J. Comput. Syst. Sci.*, 74(1):97–114, 2008.

[AR09] Jacob Abernethy and Alexander Rakhlin. Beating the adaptive bandit with high probability. In *The 22nd Annual Conference on Learning Theory (COLT 2009)*, 2009.

[BDH$^+$] Peter L. Bartlett, Varsha Dani, Thomas P. Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *The 21st Annual Conference on Learning Theory (COLT 2008)*, pages 335–342.

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

[CBMS07] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.

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

[DH06] Varsha Dani and Thomas P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *ACM-SIAM Symposium on Discrete Algorithms (SODA).*, pages 937–943. ACM Press, 2006.

[DHK07] Varsha Dani, Thomas P. Hayes, and Sham Kakade. The price of bandit information for online optimization. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2007.

[DHK08] Varsha Dani, Thomas Hayes, and Sham Kakade. The price of bandit information for online optimization. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[FKM05] Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *ACM-SIAM Symposium on Discrete Algorithms (SODA).*, pages 385–394. SIAM, 2005.

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

[HAK07] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[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.

[HK09a] Elad Hazan and Satyen Kale. Better algorithms for benign bandits. In Claire Mathieu, editor, *ACM-SIAM Symposium on Discrete Algorithms (SODA).*, pages 38–47. SIAM, 2009.

[HK09b] Elad Hazan and Satyen Kale. On stochastic and worst-case models for investing. In *Advances in Neural Information Processing Systems (NIPS) 22*, 2009.

[HK10] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: regret bounded by variation in costs. *Machine Learning*, 80(2-3):165–188, 2010.

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

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

[TW03] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 928–936. AAAI Press, 2003.

A The FTL-BTL Lemma

The following proof is essentially due to [KV05]

proof of Lemma 1. 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:

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

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

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

□