First-Order Methods in Online Convex Optimization

WONG, Hok Fong   LAM, Sze Yin
Department of Computer Science and Engineering
The Chinese University of Hong Kong
{silaswhf, 1155193130}@link.cuhk.edu.hk

Abstract

Online convex optimization deals with making sequential decisions in an uncertain environment where data arrives in a streaming fashion. Hence, it is necessary, and beneficial, to take a robust approach, by applying an optimization method that learns as more aspects of the problem are observed. This view of optimization as a process has led to successes in developing efficient algorithms for prominent online learning problems. In this report, we present the key concepts in online convex optimization, several classic first-order online convex optimization algorithms that are robust to adversarial attacks, and showcase the applicability of the framework to the online email classification task.

1   The Online Convex Optimization (OCO) Framework

Online convex optimization considers optimization as a process [1], where at round t the online optimizer chooses a vector $\mathbf{w}_t$ from a convex decision set $\mathcal{S} \subseteq \mathbb{R}^n$ in the Euclidean space. After the optimizer has committed to a choice, a convex cost function $f_t \in \mathcal{F} : \mathcal{S} \mapsto \mathbb{R}$ is revealed, which is picked from a bounded family of cost functions by the adversary. The optimizer then suffer a loss of $f_t(\mathbf{w}_t)$, which is undesired. We summarize the process of online convex optimization as the following [2].

Model 1: Online Convex Optimization

---
1 **for** $t = 1, 2, \ldots, T$ **do**
2 | Predict a vector $\mathbf{w}_t \in \mathcal{S}$
3 | Receive a convex loss function $f_t : \mathcal{S} \to \mathbb{R}$
4 | Suffer loss $f_t(\mathbf{w}_t)$

The ultimate goal of the optimizer is to minimize the cumulative sum of the losses suffered along its run. However, given that the adversary's behaviour is adaptive in nature, we have to define another fair metric called the "regret", by benchmarking a strategy with respect to the optimal one. To be precise, the regret incurred by a sequence of t decisions $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T \in \mathcal{S}$ given the revealed loss functions $f_1, f_2, \ldots, f_T$, is defined as

$$\sum_{i=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{S}} \sum_{i=1}^{T} f_t(\mathbf{w}),$$

which is the difference between the loss accumulated and the minimum loss that could have been accumulated. Therefore, we restate the optimizer's goal as having the lowest possible regret relative to a hypothesis class $\mathcal{H}$ which is

$$\text{Regret}_T(\mathcal{H}) = \max_{h^\star \in \mathcal{H}} \text{Regret}_T(h^\star).$$

We also consider that an algorithm is learning if its regret is sublinear in T, i.e. $\text{Regret}_T(\mathcal{H}) = o(T)$, implying that the regret per round asymptotically converges to zero as T tends to infinity.

Clearly, without any additional assumptions, no algorithm can achieve low regret at all as there is no correlation between past and present rounds. This impossibility result is attributed to Cover. Thus, several restrictions are necessary for this framework:

Ass. **1**: The losses determined by an adversary should not be allowed to be unbounded. Otherwise, the adversary could keep decreasing the scale of the loss at each step, and never allow the algorithm to recover from the loss of the first step. Thus, we assume that the losses lie in some bounded region.

Ass. **2**: The decision set must be bounded and/or structured, though not necessary finite. Otherwise, the adversary can assign high loss to all the strategies chosen by the player indefinitely, while setting apart some strategies with zero loss.

2 Classic Problems that Can Be Modeled via Online Convex Optimization

Online Convex Optimization (OCO) has become a leading online learning framework due to its modeling capability and its influence on the design of efficient online learning algorithms. In Online Learning, a model is always required to perform real-time adaptation to continuous data arrivals. This learning scheme also allows a model incrementally learn a concept and update accordingly without the need to retrain on the entire dataset, efficiently reduces memory and computational costs by processsing one data point at a time. In this section, we describe problems that arise in online learning and study some special cases of how they could fit into the OCO framework [1].

2.1 Online Regression / Online Classification Tasks

Online regression or classification has many applications, such as email spam classification [3]. We consider a linear classifier using a soft-margin SVM formulation for this example in the OCO framework, where the predictor receives some features of the t-th email $\mathbf{x}_t \in \mathbb{R}^d$ in round t with the "bag-of-words" representation along with some additional attributes. We also require the agent to learn a filter $\mathbf{w} \in \mathbb{R}^d$, which is norm-bounded within a certain radius. Then, the cost functions are determined by $f_t(\mathbf{w}) = \max\{0, 1 - y_t \mathbf{w}^\top \mathbf{x}_t\}$, which is a convex hinge loss function measured against the groundtruth label y_t in every round t.

As a sidenote, we used a surrogate loss function for convexification, rather than a discontinuous, non-convex 0/1 loss $f_t(\mathbf{w}) = \mathbf{1}[\text{sign}(\langle \mathbf{x}_t, \mathbf{w} \rangle) \neq y_t]$, which leads to efficient and computationally-tractable techniques for finding a solution to the optimization problem.

2.2 Expert's Advice

For the problem of prediction with expert advice, where the decision maker has to choose among the advice of d experts. Denote by $p_t \in \{1, 2, \ldots, d\}$ the chosen expert, then the learner receives a vector $\mathbf{y}_t \in [0, 1]^d$, where $y_t[i]$ is the cost of following the advice of the i-th expert. The learner pays the loss $y_t[p_t]$. The decision space in this problem is discrete.

However, by allowing the learner to randomize its predictions where we let $\mathcal{S} = \{\mathbf{w} \in \mathbb{R}^d : \mathbf{w} \geq 0 \wedge \|\mathbf{w}\|_1 = 1\}$ be the probability simplex forming a convex set, and let the learner choose $\mathbf{w}_t \in \mathcal{S}$ according to the distribution, we can restrict the power of the adversarial environment by analyzing the expected cost of the learner

$$\mathbb{E}[y_t[p_t]] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] y_t[i] = \langle \mathbf{w}_t, \mathbf{y}_t \rangle.$$

Now, in OCO, we can set $f_t(\mathbf{w}) = \langle \mathbf{w}, \mathbf{y}_t \rangle$ and analyze the regret against the set of competing vectors U, i.e. the d singletons of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$, corresponding to always following the advice of a single expert. Note that the decision set $\mathcal{S}$ is a convex simplex, and the cost functions are linear and bounded in the L_∞-norm.

3 First-Order Online Convex Optimization Algorithms

3.1 Follow-the-Leader

One primitive idea is to consider at any time the optimal vector which incurs the minimal loss on all past rounds [5]. This strategy is known as "Follow-the-Leader" (FTL) in machine learning [2, 3].

Algorithm 2: Follow-The-Leader Algorithm (Kalai and Vempala, 2005)

  1 **for** $i = 1, 2, \ldots, T$ **do**
    2   Predict a vector $\mathbf{w}_i = \arg\min_{\mathbf{w} \in \mathcal{S}} \sum_{t=1}^{i-1} f_i(\mathbf{w})$
    3   Receive a convex loss function $f_i : \mathcal{S} \to \mathbb{R}$ and suffer loss $f_i(\mathbf{w}_i)$

Although Theorem 3.1 shows a success of FTL for a sub-family of online convex optimization, FTL could fail in a worse-case scenario even for the linear optimization problem, shown in Example 3.2 [2, 3].

Theorem 3.1 (Regret of FTL on Online Quadratic Optimization) *Consider the online optimization problem where at each round $f_t(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}_t\|_2^2$ for some vector $\boldsymbol{z}_t$. Then, running FTL on the problem with $\mathcal{S} = \mathbb{R}^d$ and let $L = \max_t \|\boldsymbol{z}_t\|_2$. The regret of FTL with respect to all vectors $\boldsymbol{u} \in \mathbb{R}^d$ is at most $4L^2(\log(T) + 1)$.*

Example 3.2 (Failure of FTL) *Let $\mathcal{S} = [-1, +1]$ and conider the sequence of losses $f_t(w) = z_t w$ where*

$$z_t = \begin{cases} -0.5, & \text{if } t = 1 \\ 1, & \text{if } t \text{ is even} \\ -1, & \text{if } t > 1 \wedge t \text{ is odd} \end{cases}$$

Then, the predictions of FTL is set to $w_t = 1$ for t odd and $w_t = -1$ for t even. Then, the cumulative loss and regret will both be T as the fixed solution $u = 0 \in \mathcal{S}$ has cumulative loss 0.

We observed that FTL fails in Example 3.2 because of its unstable predictions, $\mathbf{w}_t$ shifts drastically from round to round where we only added a single loss function to the objective of the optimization problem in the definition of $\mathbf{w}_t$. This is shown in Figure 3 in Appendix A. In contrast, FTL works fine for the quadratic problem since the gradient is Lipschitz, making $\mathbf{w}_{t+1}$ close to $\mathbf{w}_t$.

3.2 Follow-the-Regularized-Leader

To stablize FTL, one way is to add a regularization term. Specifically, we consider regularizers that are strongly-convex over $\mathcal{S}$, yielding the FTRL algorithm below [2, 4].

Algorithm 3: Follow-The-Regularized-Leader Algorithm (Shalev-Shwartz, 2007)

  1 **for** $i = 1, 2, \ldots, T$ **do**
    2   Predict a vector $\mathbf{w}_i = \arg\min_{\mathbf{w} \in \mathcal{S}} \sum_{t=1}^{i-1} f_i(\mathbf{w}) + R(\mathbf{w})$
    3   Receive a convex loss function $f_i : \mathcal{S} \to \mathbb{R}$ and suffer loss $f_i(\mathbf{w}_i)$

Based on Theorem 3.3, we have an $O(\sqrt{T})$ regret bound for a general family of convex functions. The case for the online linear optimization problem is shown in Example 3.4.

Theorem 3.3 (Regret of FTRL) *Let $f_1, f_2, \ldots, f_T$ be a sequence of convex functions such that f_t is L_t-Lipschitz with respect to some norm $\|\cdot\|$. Let L be such that $\frac{1}{T}\sum_{t=1}^{T} L_t^2 \le L^2$. Then, running FTRL on the sequence with a σ-strongly-convex regularization function with respect to the same norm. For all $\boldsymbol{u} \in \mathcal{S}$, $Regret_T(\boldsymbol{u}) \le R(\boldsymbol{u}) - \min_{\boldsymbol{v} \in \mathcal{S}} R(\boldsymbol{v}) + TL^2/\sigma$. Specifically, if $U = \{\boldsymbol{u} : \|\boldsymbol{u}\|_2 \le B\}$ and $\eta = \frac{B}{L\sqrt{2T}}$ then $Regret_T(U) \le BL\sqrt{2T}$.*

Example 3.4 (Regret of FTRL on Online Linear Optimization) *Consider FTRL on the sequence of losses $f_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{z}_t \rangle$ and let $\mathcal{S} = \mathbb{R}^d, U = \{\boldsymbol{u} : \|\boldsymbol{u}\|_2 \le B\}$, $R(\boldsymbol{w}) = \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2$ where $\eta =*

$\frac{B}{L\sqrt{2T}} > 0$. *Then*, $\boldsymbol{w}_{t+1} = -\eta \sum_{i=1}^{t} \boldsymbol{z}_i$, *and* $Regret_T(\boldsymbol{u}) \leq R(\boldsymbol{u}) + TL^2\eta = \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 + TL^2\frac{B}{L\sqrt{2T}} \leq \frac{BL\sqrt{2T}}{2} + TL\frac{B\sqrt{2T}}{2T} = BL\sqrt{2T}$ *for all* $\boldsymbol{u} \in \mathcal{S}$.

3.3 Online Gradient Descent

Inspired by standard gradient descent from offline optimization, the simplest algorithm that applies to the most general setting of online convex optimization is online gradient descent (OGD) [6].

Algorithm 4: Online Gradient Descent Algorithm (Zinkevich, 2003)

---

1 Initialize and predict using the vector $\mathbf{w}_1 = \mathbf{0}$
2 for $i = 2, \ldots, T$ **do**
3 $\quad$ Predict a vector $\mathbf{w}_i = \mathbf{w}_{i-1} - \eta\mathbf{z}_{t-1}$, where $\mathbf{z}_{t-1} \in \partial f_{t-1}(\mathbf{w}_{t-1})$
4 $\quad$ Receive a convex loss function $f_i : \mathcal{S} \to \mathbb{R}$ and suffer loss $f_i(\mathbf{w}_i)$

The analysis for OGD is summarized below in Theorem 3.5.

Theorem 3.5 (Regret of OGD) *Let* $f_1, f_2, \ldots, f_T$ *be a sequence of convex functions such that* f_t *is* L_t*-Lipschitz with respect to some norm* $\|\cdot\|$. *Let* L *be such that* $\frac{1}{T}\sum_{t=1}^{T} L_t^2 \leq L^2$. *Then, the regret of OGD satisfies* $Regret_T(\boldsymbol{u}) \leq \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 + \eta TL^2$, *for all* $\boldsymbol{u} \in \mathcal{S}$. *Specifically, if* $U = \{\boldsymbol{u} : \|\boldsymbol{u}\|_2 \leq B\}$ *and* $\eta = \frac{B}{L\sqrt{2T}}$ *then* $Regret_T(U) \leq BL\sqrt{2T}$.

3.4 Online Mirror Descent

In this subsection, we derive and analyze the family of Online Mirror Descent (OMD) algorithms from the FTRL framework. Compared to FTRL, OMD has a much simpler update step, but it achieves the same regret bound as FTRL.

The starting point for OMD is to apply FTRL on a sequence of linear functions [7]. Then, assuming that $R(\mathbf{w}) = \infty$ for $\mathbf{w} \notin \mathcal{S}$ and the notation $\mathbf{z}_{1:t} = \sum_{i=1}^{t} \mathbf{z}_i$, the prediction of FTRL can be written as follows:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} R(\mathbf{w}) + \langle \mathbf{w}, \mathbf{z}_{1:t} \rangle = \arg\max_{\mathbf{w}} \langle \mathbf{w}, -\mathbf{z}_{1:t} \rangle - R(\mathbf{w}).$$

Rewriting $g(\boldsymbol{\theta}) = \arg\max_{\mathbf{w}} \langle \mathbf{w}, \boldsymbol{\theta} \rangle - R(\mathbf{w})$, we have the two-step update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$$

$$\mathbf{w}_{t+1} = g(\boldsymbol{\theta}_{t+1})$$

Now, if f_t is convex but nonlinear, then we can use subgradients of f_t at $\mathbf{w}_t$ to linearize the problem, so that $f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \langle \mathbf{w}_t, \mathbf{z}_t \rangle - \langle \mathbf{u}, \mathbf{z}_t \rangle$; hence, the regret with respect to the nonlinear loss functions is upper bounded by the regret with respect to the linear functions, yielding the OMD framework below [7].

Algorithm 5: Online Mirror Descent Algorithm (Hazan and Kale, 2008)

---

1 Initialize $\boldsymbol{\theta}_1 = \mathbf{0}$
2 for $i = 1, \ldots, T$ **do**
3 $\quad$ Predict a vector $\mathbf{w}_i = g(\boldsymbol{\theta}_t)$
4 $\quad$ Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{z}_t$ where $\mathbf{z}_t \in \partial f_t(\mathbf{w}_t)$
5 $\quad$ Receive a convex loss function $f_i : \mathcal{S} \to \mathbb{R}$ and suffer loss $f_i(\mathbf{w}_i)$

It is also clear that OGD is a special case of OMD, where $\mathcal{S} = \mathbb{R}^d$ and $g(\boldsymbol{\theta}) = \eta\boldsymbol{\theta}$ for some $\eta > 0$. For the general case where g is non-linear, the update of $\boldsymbol{\theta}$ is being carried out in a "dual" space by subtracting the gradient out of it, and the actual prediction is mirrored to the set $\mathcal{S}$ via the function g. The choice of the regularizer forms a sense of duality: transforming the space of gradient updates, from $\mathbb{R}^n$ onto itself. This is summarized in Figure 1.
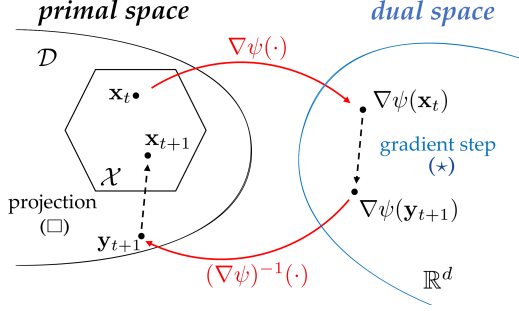
Figure 1: OMD update in terms of duality mappings, where $\nabla \psi$ is the mirror map and $(\nabla \psi)^{-1}$ is the inverse of the mirror map.

We provide a generic bound for the OMD family of algorithms based on the analysis of the FTRL algorithm, concluded by Theorem 3.6.

Theorem 3.6 *Let R be a $(1/\eta)$-strongly-convex function over $\mathcal{S}$ with respect to a norm $\|\cdot\|$. Assume that OMD is run on the sequence with a link function $g(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{w} \in \mathcal{S}}(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle - R(\boldsymbol{w}))$. Then, for all $\boldsymbol{u} \in \mathcal{S}$, $Regret_T(\boldsymbol{u}) \leq R(\boldsymbol{u}) - \min_{\boldsymbol{v} \in \mathcal{S}} R(\boldsymbol{v}) + \eta \sum_{t=1}^{T} \|\boldsymbol{z}_t\|_\star^2$, where $\|\cdot\|_\star$ is the dual norm. Furthermore, if f_t is L_t-Lipschitz with respect to $\|\cdot\|$, then $\|\boldsymbol{z}_t\|_\star \leq L_t$.*

3.5 The Doubling Trick

For the algorithms demonstrated earlier, the parameter η depends on the time horizon T, which should be avoided in real applications. However, the doubling trick enables us to convert such an algorithm requiring the knowledge of T into one without the knowledge that enjoys the same (asymptotic) regret bound. The fundamental idea is to divide the time into periods of increasing size and run the original algorithm on each period [2].

Algorithm 6: The Doubling Trick

---
1 **for** $i = 0, 1, 2, \ldots$ **do**
2 $\quad$ Run the algorithm $\mathcal{A}$ on the 2^m rounds $t = 2^m, \ldots, 2^{m+1} - 1$.

To see that the converted algorithm indeed achieves the bound, note that the regret of $\mathcal{A}$ on each period of 2^m rounds is at most $\alpha \sqrt{2^m}$. Therefore, the total regret is at most

$$\sum_{i=1}^{\lceil \log_2 T \rceil} \alpha \sqrt{2^m} = \alpha \sum_{i=1}^{\lceil \log_2 T \rceil} (\sqrt{2})^m = \alpha \frac{1 - \sqrt{2}^{\lceil \log_2 T \rceil + 1}}{1 - \sqrt{2}} \leq \alpha \frac{1 - \sqrt{2T}}{1 - \sqrt{2}} \leq \frac{\sqrt{2}}{\sqrt{2} - 1} \alpha \sqrt{T}.$$

Hence, the regret is only worse by a multiplicative factor $\frac{\sqrt{2}}{\sqrt{2}-1}$.

4 Experiment

Online Email Spam Classification We demonstrate the potential of some algorithms in Section 3 with the email categorization task, utilizing a soft-margin linear SVM with hinge loss, as has been described in Section 2.1.

Dataset. We use the `Spambase` dataset adapted from the UCI Machine Learning Repository. The dataset contains 4601 emails, each represented by 57 features $n = 57$. We assume that the emails arrives in sequence and are classified one after the other, making $T = 4601$ predictions. All the 4601 feature vectors are normalized with an L_2-norm of 1.

Algorithms and Settings. We train a model to learn a filter $\mathbf{w} \in \mathbb{R}^{4601}$, using FTL, FTRL with the regularizer $R(\mathbf{w}) = \frac{1}{2\eta}\|\mathbf{w}\|_2^2$, and OGD with the learning rate same as η. We emphasize that η is

chosen to minimize the regret bound, which is given by

$$\eta = \frac{B}{L\sqrt{2T}} = \frac{2}{1 \times \sqrt{2 \times 4601}} = 0.0208.$$

Result. We display the results for the online email classification task in Table 1.

First, the regret of FTRL is similar to that of OGD, which verifies the theoretical regret bound

$$\text{Regret}_T(U) \le BL\sqrt{2T} = 2 \times \sqrt{2 \times 4601} = 191.$$

Second, we can see that the regret's growth is limited under a line (shown in Figure 2), and seems to be asymptotically sublinear.
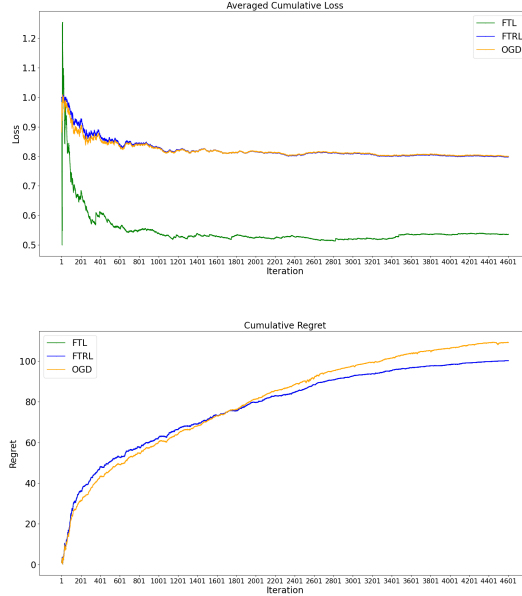

Figure 2: Performance of various algorithms in the email classfication tasks, illustrating the averaged cumulative loss for each round (upper panel) and the cumulative regret till round t (lower panel).

Thirdly, it is also surprising that FTL has outperformed both FTRL and OGD (see Table 1). This is due to the aggressive nature of FTL, we can see that the regret per round of FTL surged up to 1.2, which is vulnerable to attacks. We suspect that the dataset is lenient to FTL as the algorithm tunes the model without being tricked, which helps it achieve a better result by taking larger steps for updates, and eventually, converges to the optimal point.

Table 1: Main results for the email classification task (optimal vector is taken from U with $B = 2$, dataset shuffled for 10 independent experiments).

Methods	FTL	FTRL	OGD
Avg. mistakes	$1,074.20 \pm 37.42$	$1,814.70 \pm 3.16$	$1,818.80 \pm 42.1$
Cumulative Regret	$-1,122.78 \pm 62.53$	100.76 ± 0.86	113.48 ± 6.50

5   Summary

In this paper, we present several first-order algorithms in online convex optimization that are feasible for the online learning settings. For each of these algorithms, we also provide theoretical guarantees on the regret bound that justifies the learnability of the algorithms. We have also demonstrated one possible application of online convex optimization on the online email spam classification task, verifying the regret analyses.

References

[1] Orabona, F. (2023) A Modern Introduction to Online Learning. *arXiv*. `https://arXiv.org/pdf/1912.13213`.

[2] Shalev-Shwartz, S. (2011) Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning* **4**(2). `https://www.cs.huji.ac.il/~shais/papers/OLsurvey.pdf`.

[3] Hazan, E. (2021) Introduction to Online Convex Optimization. *arXiv*. `https://arxiv.org/pdf/1909.05207`.

[4] Shalev-Shwartz, S. (2007) *Online Learning and Online Learning: Theory, Algorithms, and Applications.* Doctoral Thesis, Hebrew University. `https://home.ttic.edu/~shai/papers/ShalevThesis07.pdf`.

[5] Kalai, A. and Vempala, S. (2005) Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* **71**(3):291-307.

[6] Zinkevich, M. (2003) Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th International Conference on Machine Learning*, 928-936.

[7] Hazan, E. and Kale, S. (2008) Extracting certainty from uncertainty: Regret bounded by variation in costs. *The 21st Annual Conference on Learning Theory (COLT)*, 57-68.

A   Failure of FTL

Consider Example 3.2 again, let $\mathcal{S} = [-1, +1]$ and the sequence of losses $f_t(w) = z_t w$ where

$$
z_t = \begin{cases} -0.5, & \text{if } t = 1 \\ 1, & \text{if } t \text{ is even} \\ -1, & \text{if } t > 1 \wedge t \text{ is odd} \end{cases}
$$

Then, the predictions of FTL is set to $w_t = 1$ for t odd and $w_t = -1$ for t even.

In Figure 3 and 4, we colored the predictions w_t green, in arrows, and the scalar z_t picked by the adversary which decides the loss function, as blue. While the algorithm wishes to receive z_t's that lie on the other side of the y-axis, the adversary does the opposite by picking exactly what FTL provides, resulting in a loss of 1 in each round.
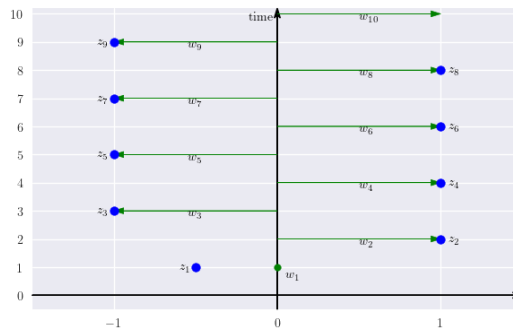

Figure 3: Illustration of the prediction of FTL and the choice made by the adversary.

In contrast, FTRL is more conservative as the step size for the online linear optimization is only η per update, adjusted against the adversarial environment.
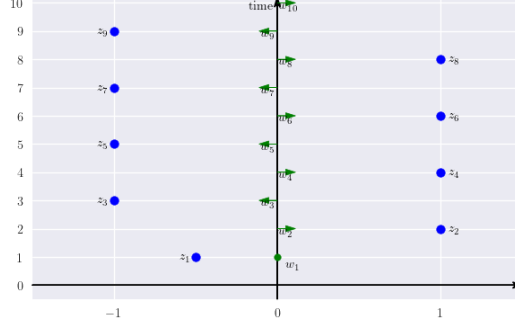
Figure 4: Illustration of the prediction of FTRL and the choice made by the adversary.

B Proof of Theorems

B.1 Proof of Theorem 3.1

Lemma 1 *Let $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$ be the sequence of vectors produced by FTL. Then, for all $\boldsymbol{u} \in S$ we have*

$$Regret_T(\boldsymbol{u}) = \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \leq \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})).$$

Proof. Proof by induction. Subtracting $\sum_t f_t(\mathbf{w}_t)$ from both sides of the inequality and rearranging yields $\sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) \leq \sum_{t=1}^{T} f_t(\mathbf{u})$. The base case is direct from the definition. Assume the inequality holds for T, then for $T+1$ we have

$$\min_{\mathbf{u} \in \mathcal{S}} \sum_{t=1}^{T+1} f_t(\mathbf{u}) = \sum_{t=1}^{T+1} f_t(\mathbf{w}_{T+2}) \overset{\text{by } def.}{\geq} \sum_{t=1}^{T} f_t(\mathbf{w}_{T+1}) + f_{T+1}(\mathbf{w}_{T+2})$$

$$\overset{\text{by IH}}{\geq} \sum_{t=1}^{T} f_t(\mathbf{w}_{t+1}) + f_{T+1}(\mathbf{w}_{T+2}) = \sum_{t=1}^{T+1} f_t(\mathbf{w}_{t+1}).$$

$\square$

Proof of Theorem 3.1. It is easy to verify that $\mathbf{w}_t = \frac{1}{t-1}\sum_{i=1}^{t-1}\mathbf{z}_i$. We can also rewrite $\mathbf{w}_{t+1} = \frac{1}{t}(\mathbf{z}_t + (t-1)\mathbf{w}_t) = (1-1/t)\mathbf{w}_t + 1/t \cdot \mathbf{z}_t$, which yields $\mathbf{w}_{t+1} - \mathbf{z}_t = (1-1/t)(\mathbf{w}_t - \mathbf{z}_t)$. Therefore,

$$f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) = \frac{1}{2}\|\mathbf{w}_t - \mathbf{z}_t\|_2^2 - \frac{1}{2}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 = \frac{1}{2}\left(1 - (1-1/t)^2\right)\|\mathbf{w}_t - \mathbf{z}_t\|_2^2 \leq \frac{1}{2}\|\mathbf{w}_t - \mathbf{z}_t\|_2^2.$$

Since $\mathbf{w}_t$ is the average of $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$ we have $\|\mathbf{w}_t\|_2 \leq L$ and therefore, $\|\mathbf{w}_t - \mathbf{z}_t\| \leq 2L$. Hence,

$$\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1})) \leq (2L)^2 \sum_{t=1}^{T} \frac{1}{t} \leq 4L^2(\log T + 1).$$

$\square$

B.2 Proof of Theorem 3.3

Proof of Theorem 3.3. Observe that running FTRL on $f_1, \ldots, f_T$ is equivalent to running FTL on $f_0, f_1, \ldots, f_T$ where $f_0 = R$. Using Lemma 1, we obtain

$$\sum_{t=0}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{u})) \leq \sum_{t=0}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1})).$$

Rearranging and use $f_0 = R$, we now have

$$\text{Regret}_T(\mathbf{u}) \leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1})) \leq R(\mathbf{u}) - \min_{\mathbf{v} \in \mathcal{S}} R(\mathbf{v}) + \sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1})).$$

Since f_t is L_t-Lipschitz with respect to a norm, $f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq L\|\mathbf{w}_t - \mathbf{w}_{t+1}\|$. By strong-convexity of R, we consider $F_t(\mathbf{w}) = \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$ which is also σ-strongly-convex, and note that $\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{S}} F_t(\mathbf{w})$. This implies

$$F_t(\mathbf{w}_{t+1}) \geq F_t(\mathbf{w}_t) + \frac{\sigma}{2}\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2.$$

Repeating the same argument for F_{t+1} and its minimizer $\mathbf{w}_{t+1}$ yields

$$F_{t+1}(\mathbf{w}_t) \geq F_{t+1}(\mathbf{w}_{t+1}) + \frac{\sigma}{2}\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2.$$

Hence, summing up the two inequalities we have

$$\sigma\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2 \leq f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}) \leq L_t\|\mathbf{w}_t - \mathbf{w}_{t+1}\|.$$

Rearranging, we obtain $\|\mathbf{w}_t - \mathbf{w}_{t+1}\| \leq L_t/\sigma$, which concludes the proof. $\qquad\square$

B.3 Proof of Theorem 3.5

Proof of Theorem 3.5. In each round t, there exists a subgradient $\mathbf{z}_t$ such that for all $\mathbf{u}$, $f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle$. Hence,

$$\sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{u})) \leq \sum_{t=1}^{T}(\langle \mathbf{w}_t, \mathbf{z}_t \rangle - \langle \mathbf{u}, \mathbf{z}_t \rangle).$$

Also,

$$\text{Regret}_T(\mathbf{u}) \leq R(\mathbf{u}) - R(\mathbf{w}_1) + \sum_{t=1}^{T}(f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t+1}))$$

$$\leq \frac{1}{2\eta}\|\mathbf{u}\|_2^2 + \sum_{t=1}^{T}\langle \mathbf{w}_t - \mathbf{w}_{t+1}, \mathbf{z}_t \rangle = \frac{1}{2\eta}\|\mathbf{u}\|_2^2 + \eta\sum_{t=1}^{T}\|\mathbf{z}_t\|_2^2.$$

Since f_t is Lipschitz, the term $\sum_{t=1}^{T}\|\mathbf{z}_t\|_2^2$ can be bounded by $\sum_{t=1}^{T} L_t^2 \leq L$, concluding the proof. $\qquad\square$

B.4 Proof of Theorem 3.6

Proof of Theorem 3.6. As shown previously, $f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle$. Since OMD is equivalent to running FTRL on the sequence of linear functions with the regularization $R(\mathbf{w})$, it immediately follows the theorem. $\qquad\square$