

Online Convex Optimization

1 Introduction

In the previous lecture, an Online Convex Optimization(**OCO**) algorithm, Follow The Leader(**FTL**) was introduced. In this lecture, we will continue our study with the **FTL** and look at some other **OCO** algorithms and analyze their regret bounds.

1.1 Quick Revision

From the previous lecture, the **FTL** algorithm is given as in Algorithm 1

Algorithm 1: Follow The Leader(**FTL**)

```

1: for  $t = 1, 2, \dots$ , do
2:   Predict  $\mathbf{x}^t = \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{\tau=1}^{t-1} \ell^\tau(\mathbf{x})$ 
3:   Receive  $\ell^t$ 
4: end for

```

It was also proved that the regret of **FTL** is upper bounded by the cumulative difference between the loss of $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$ given in lemma 10.1 Shalev-Shwartz (2012)

Lemma 10.1. $\text{Regret}_T(\mathbf{x}) = \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x})) \leq \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1})) \quad \forall \mathbf{x} \in \mathcal{C}$

Example 10.1. Consider a **FTL** algorithm with loss function defined as $\ell^t(x) = \{x, 1-x\}$. The action taken at each time instant is either $x = 0/1$. Corresponding to each action, a loss is incurred.

Time	Action	ℓ_t	Loss
t=1	0	x	0
t=2	0	1-x	1
t=3	0	1-x	1
t=4	1	x	1

Table 1: **FTL** algorithm

At time $t = 1$ the **FTL** starts with action 0 and the adversary sends x . So, the loss incurred is 0. At $t = 2$, the **FTL** with previous loss knowledge will choose 0 but now the adversary

having earlier sent x will now send $1 - x$ and thus the loss incurred will be 1. For $t = 3, \dots, T$, the steps follows subsequently as given in table 10.1. Hence, the cumulative loss incurred is $T - 1$.

2 The Fireman Problem

Consider a city with radius R which is monitored by the fire department that deploys the fire vehicle close to the expected site of fire. This is a hypothetical case considered where the chances of fire breaking at some place at each night is high. Let, the location of the deployed vehicle is $\mathbf{x}$ on a certain day and let the location of the exact fire breaking point be $\mathbf{z}^t$. So, the loss function incurred is given as $\ell^t : \mathbf{x} \mapsto \|\mathbf{x} - \mathbf{z}^t\|_2^2$

2.1 Analysis using FTL Algorithm

Applying the **FTL** algorithm we can predict the optimal $\mathbf{x} \in \mathcal{C}$ as

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{\tau=1}^t \|\mathbf{x} - \mathbf{z}^\tau\|_2^2$$

Hence, the optimal $\mathbf{x}$ is found by setting the gradient of the objective problem to zero. Thus,

$$\begin{aligned} \mathbf{x}^{t+1} &= \frac{1}{t} \sum_{\tau=1}^t \mathbf{z}^\tau \\ &= \frac{1}{t} \{\mathbf{z}^t + (t-1)\mathbf{x}^t\} \end{aligned}$$

So, subtracting $\mathbf{z}^t$ from the above equation we get

$$\begin{aligned} \mathbf{x}^{t+1} - \mathbf{z}^t &= \left(\frac{1}{t} - 1\right)\mathbf{z}^t + \left(1 - \frac{1}{t}\right)\mathbf{x}^t \\ &= \left(1 - \frac{1}{t}\right)(\mathbf{x}^t - \mathbf{z}^t) \end{aligned}$$

Now, considering the loss function as defined in the problem we get,

$$\begin{aligned} \ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1}) &= \left[1 - \left(1 - \frac{1}{t}\right)^2\right] \|\mathbf{x}^t - \mathbf{z}^t\|_2^2 \\ &\leq \frac{2}{t} \times \|\mathbf{x}^t - \mathbf{z}^t\|_2^2 \end{aligned}$$

Hence, the forward regret is given as

$$\begin{aligned} \text{Regret}_T(\mathbf{FTL}) &= \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1})) \\ &\leq 2 \times \sum_{t=1}^T \frac{\|\mathbf{x}^t - \mathbf{z}^t\|_2^2}{t} \\ &\leq 8R^2 \log T \end{aligned}$$

where, $\max(\|\mathbf{x}^t - \mathbf{z}^t\|) = 2R$

Corollary 10.2. The regret for running an **FTL** on an Online Quadratic Optimization problem is atmost $8R^2 \log T$.

3 Follow The Regularized Leader(**FTRL**)

The **FTL** algorithm is not stable but can be stabilized by adding a regularization term. Different regularization function will give different algorithms with different regret bounds Shalev-Shwartz (2012). The more stable predictions of an algorithm, lower is the regret. Considering $R(\mathbf{x})$ to be the regularized parameter that makes the loss function strictly convex, the **FTRL** algorithm is given below in Algorithm 2.

Algorithm 2: Follow The Regularized Leader(**FTRL**)

```

1: for  $t = 1, 2, \dots$ , do
2:   Predict  $\mathbf{x}^t = \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{\tau=1}^{t-1} \ell^\tau(\mathbf{x}) + R(\mathbf{x})$ 
3:   Receive  $\ell^t$ 
4: end for

```

Lemma 10.3. Let, $\mathbf{x}^1, \mathbf{x}^2, \dots$ be the predictions for $t = 1, 2, \dots$ for the **FTRL** Shalev-Shwartz (2012). Thus, $\forall \mathbf{x} \in \mathcal{C}$

$$\text{Regret}_T(\mathbf{FTRL}) = \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x})) \leq \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1})) + R(\mathbf{x}) - R(\mathbf{x}^1)$$

Proof : Lemma 10.3 can be easily proved by using lemma 10.1 by taking the summation from $t = 0$ to T and considering $\ell^0(\mathbf{x}) = R(\mathbf{x})$. Shalev-Shwartz (2012)

Example 10.2. Consider the loss function given as $\ell^t(\mathbf{x}) = \langle \mathbf{z}^t, \mathbf{x} \rangle$ and the regularization term be $R(\mathbf{x}) = \frac{1}{2\eta} \|\mathbf{x}\|_2^2$, $\eta > 0$. Applying the **FTRL** algorithm we get

$$\begin{aligned}
\mathbf{x}^{t+1} &= \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{\tau=1}^t \ell^\tau(\mathbf{x}) + R(\mathbf{x}) \\
&= \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{\tau=1}^t \langle \mathbf{z}^\tau, \mathbf{x} \rangle + \frac{1}{2\eta} \|\mathbf{x}\|_2^2
\end{aligned}$$

So, the differential of the above cumulative loss function gives

$$\mathbf{x}^{t+1} = -\eta \sum_{\tau=1}^t \mathbf{z}^\tau = \mathbf{x}^t - \eta \mathbf{z}^t. \quad // \quad \text{where, } (\mathbf{x}^t = -\eta \sum_{\tau=1}^{t-1} \mathbf{z}^\tau)$$

Now, the cumulative difference between the loss at $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$ will be

$$\begin{aligned}
\sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1})) &= \sum_{t=1}^T (\langle \mathbf{z}^t, \mathbf{x}^t \rangle - \langle \mathbf{z}^t, \mathbf{x}^{t+1} \rangle) \\
&= \sum_{t=1}^T \langle \mathbf{z}^t, \mathbf{x}^t - \mathbf{x}^{t+1} \rangle \\
&= \sum_{t=1}^T \langle \mathbf{z}^t, \eta \mathbf{z}^t \rangle \\
&= \eta \sum_{t=1}^T \|\mathbf{z}^t\|_2^2
\end{aligned}$$

Hence, by using lemma 10.3 we get,

$$\begin{aligned}
\text{Regret}_T(\mathbf{FTRL}) &\leq \eta \sum_{t=1}^T \|\mathbf{z}^t\|_2^2 + \frac{1}{2\eta} \|\mathbf{x}^*\|_2^2 - R(\mathbf{x}^1) \\
&\leq \eta \sum_{t=1}^T \|\mathbf{z}^t\|_2^2 + \frac{1}{2\eta} \|\mathbf{x}^*\|_2^2 \quad // \text{where, } R(\mathbf{x}^1) > 0 \\
&\leq \eta TG^2 + \frac{1}{2\eta} \|\mathbf{x}^*\|_2^2 \quad // \text{Assuming, } \|\mathbf{z}^t\|_2^2 \leq G^2 \\
&= \sqrt{2TG} \|\mathbf{x}^*\| \quad // \text{For, } \eta = \frac{\|\mathbf{x}^*\|}{\sqrt{2TG}}
\end{aligned}$$

3.1 Stability Analysis for FTRL Algorithm

Consider, $L_0(\mathbf{x}) = \sum_{\tau=1}^t \ell^\tau(\mathbf{x})$ and $R(\mathbf{x})$ be the regularization function, The prediction at t & $t+1$ will be given as below:

$$\mathbf{x}^t = \arg \min_{\mathbf{x} \in \mathcal{C}} L_0(\mathbf{x}) + R(\mathbf{x}) \quad (1)$$

$$\mathbf{x}^{t+1} = \arg \min_{\mathbf{x} \in \mathcal{C}} L_0(\mathbf{x}) + \ell^t(\mathbf{x}) + R(\mathbf{x}) \quad (2)$$

Let, $F^t(\mathbf{x}) = L_0(\mathbf{x}) + R(\mathbf{x})$ and $F^{t+1}(\mathbf{x}) = L_0(\mathbf{x}) + \ell^t(\mathbf{x}) + R(\mathbf{x})$ which are α -strongly convex.

Thus, by first order convexity we can express $F^t(\mathbf{x}^t)$ as

$$F^t(\mathbf{x}^t) \geq F^t(\mathbf{x}^{t+1}) + \langle \nabla F^t(\mathbf{x}^{t+1}), \mathbf{x}^t - \mathbf{x}^{t+1} \rangle + \frac{\alpha}{2} \Delta^2$$

where, $\Delta = \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2$.

Writing the above equation other way round we get,

$$F^t(\mathbf{x}^{t+1}) \geq F^t(\mathbf{x}^t) + \langle \nabla F^t(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t \rangle + \frac{\alpha}{2} \Delta^2$$

Since, $\mathbf{x}^t$ is a solution of eq. (1), thus $\nabla F^t(\mathbf{x}^t) = 0$, which gives us

$$F^t(\mathbf{x}^{t+1}) \geq F^t(\mathbf{x}^t) + \frac{\alpha}{2} \Delta^2 \quad (3)$$

Similarly, for F^{t+1} we have

$$F^{t+1}(\mathbf{x}^t) \geq F^{t+1}(\mathbf{x}^{t+1}) + \frac{\alpha}{2} \Delta^2 \quad (4)$$

Adding the two inequalities in (3) and (4) and simplifying the expression we get,

$$\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1}) \geq \alpha \Delta^2 = \alpha \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2 \quad (5)$$

If the loss functions being used are L-Lipschitz, we have

$$\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^{t+1}) \leq L \times \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2 \quad (6)$$

Combining equations (5) and (6) we get,

$$\|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2 \leq 2L/\alpha$$

Using Lipschitzness again from eq. (6) gives us

$$\ell_t(\mathbf{x}^t) - \ell_t(\mathbf{x}^{t+1}) \leq 2L^2/\alpha$$

Summing over time steps tells us that the forward regret is bounded by $2L^2/\alpha \times T$.

Setting $\alpha = \sqrt{T}$ gives us a sublinear regret bound for **FTRL**. However, notice that the same setting of α also gives us

$$\|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2 \leq 2L/\sqrt{T}$$

This explicates the stability of **FTRL**: with Lipschitz loss functions and a strong enough regularization, **FTRL** makes provably stable updates. The successive updates, $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$, are never far from each other. Notice that the above process works even if the loss functions are non-convex : Lipschitzness is all that is required. However, if the loss function is non-convex and non-Lipschitz, even then **FTRL** can be made to incur linear regret.

4 Online Gradient Descent (OGD)

The **OGD** is similar to the standard gradient descent algorithm as defined in offline framework. The **OGD** algorithm is represented in algorithm 3. At each iteration the algorithm takes a step along the direction of the gradient of the previous point. The updates take linear time given the gradient. Hazan (2015)

Algorithm 3: Online Gradient Descent (**OGD**)

```

1: for  $t = 1, 2, \dots$ , do
2:   Predict  $\mathbf{x}^t \in \mathcal{C}$ 
3:   Receive  $\ell^t$ 
4:    $\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \times \nabla \ell^t(\mathbf{x}^t)$ 
5: end for
```

Lemma 10.4. With step size, $\eta_t = \frac{D}{G\sqrt{t}}$, $t = 1, 2 \dots T$, the regret bound for **OGD** algorithm is as below Hazan (2015)

$$\text{Regret}_T = \sum_{t=1}^T \ell^t(\mathbf{x}^t) - \min_{\mathbf{x}^* \in \mathcal{C}} \sum_{t=1}^T \ell^t(\mathbf{x}^*) \leq 3GD\sqrt{T}$$

Proof: Hazan (2015) Let, $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{t=1}^T \ell^t(\mathbf{x})$. By convexity we get,

$$\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^*) \leq \nabla_t^T(\mathbf{x}^t - \mathbf{x}^*), \quad \text{where, } \nabla_t = \nabla(\ell^t(\mathbf{x}^t)) \quad (7)$$

Now, using the update rule for $\mathbf{x}^{t+1}$ we get,

$$\begin{aligned}\|\mathbf{x}^{t+1} - \mathbf{x}^*\|^2 &= \|\Pi_{\mathcal{C}}[\mathbf{x}^t - \eta_t \nabla_t] - \mathbf{x}^*\|^2 \\ &\leq \|\mathbf{x}^t - \eta_t \nabla_t - \mathbf{x}^*\|^2 \\ &= \|\mathbf{x}^t - \mathbf{x}^*\|^2 + \eta_t^2 \|\nabla_t\|^2 - 2\eta_t \nabla_t^T (\mathbf{x}^t - \mathbf{x}^*)\end{aligned}$$

Hence, from the above inequality we get,

$$2\nabla_t^T (\mathbf{x}^t - \mathbf{x}^*) = \frac{\|\mathbf{x}^t - \mathbf{x}^*\|^2 - \|\mathbf{x}^{t+1} - \mathbf{x}^*\|^2}{\eta_t} + \eta_t G^2 \quad (8)$$

where, $\|\nabla(\ell(\mathbf{x}))\| \leq G$, $\forall \mathbf{x} \in \mathcal{C}$ and $\Pi_{\mathcal{C}}$ is the projection oracle.

Combining equations (7) and (8) and finding the cumulative sum we get,

$$\begin{aligned}2 \sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^*)) &\leq 2 \sum_{t=1}^T \frac{\|\mathbf{x}^t - \mathbf{x}^*\|^2 - \|\mathbf{x}^{t+1} - \mathbf{x}^*\|^2}{\eta_t} + G^2 \sum_{t=1}^T \eta_t \\ &\leq \sum_{t=1}^T \|\mathbf{x}^t - \mathbf{x}^*\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + G^2 \sum_{t=1}^T \eta_t \quad // \text{ where, } \frac{1}{\eta_t} \triangleq 0, \|\mathbf{x}^{T+1} - \mathbf{x}^*\|^2 \geq 0 \\ &\leq D^2 \sum_{t=1}^T \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + G^2 \sum_{t=1}^T \eta_t \quad // \text{ where, } \|\mathbf{x}^t - \mathbf{x}^*\| \leq D \\ &\leq \frac{D^2}{\eta_T} + G^2 \sum_{t=1}^T \eta_t\end{aligned}$$

So, for $\eta_t = \frac{D}{G\sqrt{t}}$ the inequality becomes

$$\sum_{t=1}^T (\ell^t(\mathbf{x}^t) - \ell^t(\mathbf{x}^*)) \leq 3DG\sqrt{T} \quad (9)$$

where, $\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$.

4.1 FTRL-OGD Link

Consider the **FTRL** algorithm with the gradients of loss function seen till $1, 2, \dots, t$. So, $\mathbf{x}^{t+1} = \mathbf{FTRL}_{\eta}(\nabla \ell^1, \nabla \ell^2, \dots, \nabla \ell^t)$ is the update step of **FTRL** with η -regularizer. Hence, for any linear loss function, the step at $t+1$ gives the update step similar to the **OGD** method.

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta(\nabla \ell^t(\mathbf{x}^t)) \quad (10)$$

The above conclusion is similar to the result obtained in example 10.2 where the loss function was considered to be linear with $(\nabla \ell^t(\mathbf{x}^t)) = \mathbf{z}^t$.

Corollary 10.5. Thus, the regret bound for the **OGD** procedure can be found by the **FTRL** bound as obtained in example 10.2 i.e.,

$$\text{Regret}_T(\mathbf{OGD}) \leq G\|\mathbf{x}^*\|\sqrt{2T}$$

4.2 FTRL-Perceptron Link

Consider, the hinge-loss function given as

$$\ell^t(\mathbf{x}) = \ell^h(\langle \mathbf{z}, \mathbf{x} \rangle, y^t)$$

where, $\ell^h(\langle \mathbf{z}, \mathbf{x} \rangle, y^t) = [1 - y^t(\mathbf{z}^T \cdot \mathbf{x})]_+$

Now, consider the hypothesis set of separating hyperplanes, Shalev-Shwartz and Singer (2007)

$$\mathcal{H} = \{h_{\mathbf{z}}(\mathbf{x}) = \text{sign}(\langle \mathbf{z}, \mathbf{x} \rangle) | \mathbf{z} \in \mathbb{R}^n\}$$

Now, a 0 – 1 loss function is defined as

$$\ell_{0-1}(h_{\mathbf{z}}, (\mathbf{x}, y)) = \begin{cases} 1 & \text{if } h_{\mathbf{z}}(\mathbf{x}) \neq y \\ 0 & \text{if } h_{\mathbf{z}}(\mathbf{x}) = y \end{cases}$$

Therefore, the cumulative 0-1 loss of the online learning algorithm is the number of prediction mistakes the online algorithm makes along its run. Also, it can be easily shown that Shalev-Shwartz and Singer (2007)

$$\ell^h(\langle \mathbf{z}, \mathbf{x} \rangle, y) \geq \ell_{0-1}(h_{\mathbf{z}}, (\mathbf{x}, y))$$

Therefore, for any $\mathbf{x}^* \in \mathcal{C}$, we have

$$\begin{aligned} \text{Regret}(\mathbf{x}^*, T) &= \sum_{t=1}^T \left(\ell^h(\langle \mathbf{z}^t, \mathbf{x}^t \rangle, y^t) - \ell^h(\langle \mathbf{x}^*, \mathbf{x}^t \rangle, y^t) \right) \\ &\leq \sum_{t=1}^T \left(\ell_{0-1}(h_{\mathbf{z}^t}, (\mathbf{x}^t, y^t)) - \ell^h(\langle \mathbf{x}^*, \mathbf{x}^t \rangle, y^t) \right) \end{aligned}$$

From the above equation we get

$$\sum_{t=1}^T \ell_{0-1}(h_{\mathbf{z}^t}, (\mathbf{x}^t, y^t)) \leq \sum_{t=1}^T \ell^h(\langle \mathbf{x}^*, \mathbf{x}^t \rangle, y^t) + \text{Regret}(\mathbf{x}^*, T) \quad (11)$$

Let, $\mathcal{M}$ be the set of rounds in which $h_{\mathbf{z}^t}(\mathbf{x}^t) \neq y^t$. Hence, the L.H.S of equation (11) is $|\mathcal{M}|$. Thus, the above equation can be written as

$$\begin{aligned} |\mathcal{M}| &\leq \sum_{t \in |\mathcal{M}|} \ell^h(\langle \mathbf{x}^*, \mathbf{x}^t \rangle, y^t) + \text{Regret}(\mathbf{x}^*, |\mathcal{M}|) \\ &\leq L^h(\mathbf{x}^*, |\mathcal{M}|) + B\|\mathbf{x}^*\|\sqrt{|\mathcal{M}|} \quad // \text{ where, } L^h(\mathbf{x}^*, |\mathcal{M}|) = \sum_{t \in |\mathcal{M}|} \ell^h(\langle \mathbf{x}^*, \mathbf{x}^t \rangle, y^t) \end{aligned}$$

From the above inequality we can write Shalev-Shwartz and Singer (2007),

$$|\mathcal{M}| \leq L^h(\mathbf{x}^*, |\mathcal{M}|) + B\|\mathbf{x}^*\|\sqrt{L^h(\mathbf{x}^*, |\mathcal{M}|)} + B^2\|\mathbf{x}^*\|^2$$

where, $B = \max_t \|\mathbf{x}^t\|$. The above bound is called relative mistake bound and is the best relative mistake bound known for the Perceptron algorithm.

Corollary 10.6. The hinge loss function can be used to obtain the Perceptron’s prediction from the **OGD** algorithm.

Proof: Considering the hinge loss function as $\ell^t(x) = \ell^h(\theta^t.x, y^t) = [1 - y^t(\theta^t.x)]$.

Hence, $\nabla(\ell^t(x)) = -y^t\theta^t$.

Now, the update step of the **OGD** gives

$$x^{t+1} = x^t - \eta y^t \theta^t$$

Which is similar to the Perceptron’s method.

References

Elad Hazan. Introduction to online convex optimization. *Book draft*, 2015.

Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends[®] in Machine Learning*, 4(2):107–194, 2012.

Shai Shalev-Shwartz and Yoram Singer. Online learning: Theory, algorithms, and applications. 2007.