Online Learning: Online Convex Optimization with Full-Information

Daniel Khashabi

Summer 2014
Last Update: October 20, 2016

1 Introduction

[TODO]

2 Online Regret Minimization

Online Learning is a popular setting learning. Let's review the definition of Regret in Equation **??** for a more general setting. We define $z_t = (x_t, y_t)$:

$$R(T) = \frac{1}{T}\sum_{t=1}^{T} l(f(x_t), z_t) - \inf_{f^* \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} l(f^*(x_t), z_t) \tag{1}$$

This is the average loss of the learner, compared to the best predictor from $\mathcal{F}$.

Here is the way an online learning usually modeled

At each time step $t = 1$ to T:

- Learner (Player) chooses $f(x_t)$, for some $f \in \mathcal{F}$.
- Nature (Adversary) choose y_t.
- Learner (Player) suffers loss $l(f(x_t), z_t)$.
- Both observe the feedback of loss.

Here we consider two different scenarios:

1. Non-adaptive adversary: When the target predictions $(y_1, \ldots, y_T)$ are fixed a-priori to the learning.
2. Adaptive adversary: When each target prediction y_t is decided after observing learner's (player's) decisions up to time $t - 1$.

Usually in Machine Learning problems, we are dealing with the non-adaptive case, although the adaptive adversary also appears sometimes. Then we will need to use a little Game Theory, along with optimization.

From now on suppose our decision making function which we use in the definition of the regret in Equation 1, $f(x_t)$ is characterized by some vector of parameters $\mathbf{w}_t \in \mathcal{W}$. In other words, it is better to represent the decision making function with $f(x_t; \mathbf{w}_t)$, although for brevity we will continue using $f(x_t)$.

2.1 Follow-The-Leader (FTL)

Let us start with a greedy approach, by choosing the best decision-maker, in terms of all observation seen by now. This is commonly called Follow-The-Leader(FTL):

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose any:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{\beta} \sum_{t=1}^{t} l(f(x_t), z_t)$$

In the above setting the denominator β has no role, in the selection of $\mathbf{w}_{t+1}$, but to have coherent notation with the future formulation, we keep it in here.

2.2 Follow-The-Regularized-Leader (FTRL)

Our example prediction in the introduction is a FTL predictor. As Preposition **??** showed, there can be sequences for which this decision maker is arbitrarily bad. Let's consider the case where we add a regularization term $R(\mathbf{w})$ to our decision making, to soften our decision making under uncertainty.

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose any:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W}} \left[\frac{1}{\beta} \sum_{t=1}^{t} l(f(x_t), z_t) + R(\mathbf{w}) \right]$$

2.2.1 Slight change of notation

Let's change the notation slightly. We will use this notation later in other algorithms. Define

$$\phi_0 = R(\mathbf{w})$$

And for any $t > 0$

$$\phi_t(\mathbf{w}) = \phi_{t-1}(\mathbf{w}) + \frac{1}{\beta} l(f(x_t), z_t) \tag{2}$$

Now we can change the FTRL updates in the following form:

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose any:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W}} \phi_t(\mathbf{w})$$

One unclear point in the above notation is an efficient way of choosing $\mathbf{w} \in \mathcal{W}$. In the next section we continue change the notation and later introduce a projection based on the Bregman divergence.

2.2.2 Projections with Bregman divergence

Here we continue changing the previous notation based on the Bregman divergence. But before anything let's review Bregman divergence.

Definition 1 (Bregman Divergence)**.** *Given a function strictly-convex function ϕ, Bregman divergence divergence is defined as following:*

$$D_\phi(p, q) \triangleq \phi(p) - \phi(q) - \nabla\phi(q)^\top (p - q)$$

Here are a few properties for Bregman divergence:

- $D_\phi(p, q) \geq 0$
- *It is linear with respect to ϕ:*

$$D_{\phi_1 + \phi_2}(p, q) = D_{\phi_1}(p, q) + D_{\phi_2}(p, q)$$

- *Derivative of the Bregman divergence:*

$$\nabla_p D_\phi(p, q) = \nabla_p \phi(p) - \nabla_p \phi(q) \tag{3}$$

- *Projection with Bregman divergence onto a convex set Ω, always exists and has a unique answer:*

$$P_\Omega(q) = \arg\min_{p \in \Omega} D_\phi(p, q)$$

  Given the result in Equation 3, we have:

$$\nabla_p \phi\left(p\right)\big|_{p = P_\Omega(q)} = \nabla_p \phi\left(p\right)\big|_{p = q} \tag{4}$$

  Note that, if $\Omega = \mathbb{R}^n$, then

$$P_\Omega\left(q\right) = q.$$

- *Generalized Pythagorean theorem: For any convex set Ω, q and q, we have:*

$$D_\phi(p, q) \geq D_\phi(p, P_\Omega(q)) + D_\phi(P_\Omega(q), q)$$

- *Denote the dual (Legendre-conjugate) of ϕ with ψ. Then we have:*

$$\nabla \psi = (\nabla \phi)^{-1}$$

and

$$D_\phi(p,q) = D_\psi(\nabla\phi(q), \nabla\phi(p))$$

With this definition, we continue the change of notation, and rewrite the updates of the online learning based on Bregman divergence.

Lemma 1 (Online Learning based Bregman divergence)**.** *Suppose the loss $l(.)$ and the regularizer $R(\mathbf{w})$ are convex and unique minimum. Also suppose $\mathcal{W}$ is a convex subset of $\mathbb{R}^n$. The we have the following equality:*

$$\begin{aligned}&\arg\min_{\mathbf{w}\in\mathcal{W}} \left[\frac{1}{\beta} \sum_{t=1}^{t} l(f(x_t), z_t) + R(\mathbf{w}) \right] \\ &\qquad = \arg\min_{\mathbf{w}\in\mathcal{W}} \left[\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) \right]\end{aligned}$$

Proof. We just need to prove that, the minimizer of two expressions are the same. For a moment, suppose both of them have a unique finite minimum. We just need to take derivative with respect to the variable $\mathbf{w}$ and prove that, they have the same minimizer. From Equation 2 we know:

$$\frac{1}{\beta} l(f(x_t), z_t) = \phi_t(\mathbf{w}) - \phi_{t-1}(\mathbf{w})$$

Now adding $D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}))$ to both sides:

$$\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) = \phi_t(\mathbf{w}) - \phi_{t-1}(\mathbf{w}) + D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) \tag{5}$$

And from Equation 3 we know that:

$$\nabla_{\mathbf{w}} D_{\phi_{i-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) = \nabla_{\mathbf{w}}\phi_{i-1}(\mathbf{w}) - \nabla_{\mathbf{w}}\phi_{i-1}(P_{\mathcal{W}}(\mathbf{w}_t)) \tag{6}$$

Replacing Equation 6 into Equation 5:

$$\begin{aligned}\nabla \left[\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{i-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) \right] &= \nabla_{\mathbf{w}}\phi_t(\mathbf{w}) - \nabla_{\mathbf{w}}\phi_{t-1}(\mathbf{w}) + \nabla_{\mathbf{w}} D_{\phi_{i-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) \\ &= \nabla_{\mathbf{w}}\phi_t(\mathbf{w}) - \nabla_{\mathbf{w}}\phi_{t-1}(\mathbf{w}) + \nabla_{\mathbf{w}}\phi_{t-1}(\mathbf{w}) - \nabla_{\mathbf{w}}\phi_{t-1}(P_{\mathcal{W}}(\mathbf{w}_t)) \\ &= \nabla_{\mathbf{w}}\phi_t(\mathbf{w}) - \nabla_{\mathbf{w}}\phi_{t-1}(P_{\mathcal{W}}(\mathbf{w}_t))\end{aligned}$$

If $\nabla_{\mathbf{w}}\phi_{t-1}(P_{\mathcal{W}}(\mathbf{w}_t)) = 0$ (why?), then:

$$\nabla \left[\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{i-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t)) \right] = \nabla_{\mathbf{w}}\phi_t(\mathbf{w})$$

which, using the Equation 2 proves the desired result in this lemma. □

Using the above lemma we can rewrite the online learning in the following form:

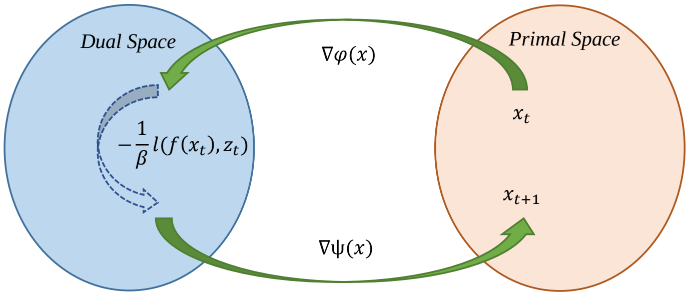

Figure 1: One can interpret using Bregman as taking the objective function into dual space, and bringing it back to primal, by a final constrained projection.

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose any:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \left[\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}_t))\right]$$

Instead of working with the above constrained optimization, it is common to split it into two parts, like the following two steps. This form of online learning is called *lazy online learning* for obvious reason.

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose with the following two steps:

1. Step forward:

$$\bar{\mathbf{w}}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \left[\frac{1}{\beta} l(f(x_t), z_t) + D_{\phi_{t-1}}(\mathbf{w}, P_{\mathcal{W}}(\mathbf{w}))\right]$$

2. Projection:

$$\mathbf{w}_{t+1} = \Pi_{\phi_t,\mathcal{W}}(\bar{\mathbf{w}}_{t+1}),$$

defined as:

$$\Pi_{\phi_t,\mathcal{W}}(\bar{\mathbf{w}}_{t+1}) \triangleq \arg\min_{\mathbf{w}\in\mathcal{W}} D_{\phi_t}(\mathbf{w}, \bar{\mathbf{w}}_{t+1}).$$

In the future sections, we will analyze different properties of our settings, with different loss functions, regulizers and other settings.

2.3 Linear loss functions

Suppose we have a linear loss function. This case is interesting for different obvious reasons:

- Linear function is special case of a convex function.
- Other functions could be approximated or bounded with linear functions
- etc.

Suppose we decompose the cost function, as linear combination of smaller cost functions represented as $\mathbf{g}(x_t, z_t)$:

$$l(f(x_t), z_t) = \mathbf{w}^\top \mathbf{g}(x_t, z_t)$$

For simplicity we denoted $\mathbf{g}(x_t, z_t)$ with $\mathbf{g}_t$. How does this change our formulation in the previous section? The difference appears because the Bregman divergence divergence doesn't change by adding a linear term to the function it was acting on, i.e. :

$$D_R(.) = D_{\phi_0}(.) = D_{\phi_1}(.) = \ldots = D_{\phi_t}(.)$$

By this we can show the following lemma:

Lemma 2 (Updates of online learning in the linear case)**.** *In online learning with linear loss, we can simplify the update in Equation* **??** *to the following form:*

$$\bar{\mathbf{w}}_{t+1} = \nabla R^* \left(\nabla R(\bar{\mathbf{w}}_t) - \frac{1}{\beta} \mathbf{g}_t \right)$$

For $t = 1$ choose $\mathbf{w}_1 \in \mathcal{W}$ randomly. For any $t \geq 1$ choose with the following two steps:

1. Step forward:
$$\bar{\mathbf{w}}_{t+1} = \nabla R^* \left(\nabla R(\bar{\mathbf{w}}_t) - \frac{1}{\beta} \mathbf{g}_t \right)$$
2. Projection:
$$\mathbf{w}_{t+1} = \Pi_{\phi_t, \mathcal{W}}(\bar{\mathbf{w}}_{t+1}),$$
defined as:
$$\Pi_{\phi_t, \mathcal{W}}(\bar{\mathbf{w}}_{t+1}) \triangleq \arg\min_{\mathbf{w} \in \mathcal{W}} D_{\phi_t}(\mathbf{w}, \bar{\mathbf{w}}_{t+1}).$$

This form of updates is commonly known as *Mirror Descent* updates, mostly due to [**?**].

2.4 Some background notes

Definition 2 (Strongly convex function)**.** *A function $R : S \to \mathbb{R}$ is called β-strongly convex over domain S, with respect to the norm $\|.\|$, if for any $\mathbf{w} \in S$:*

$$R(\mathbf{w}') \geq R(\mathbf{w}) + \langle \mathbf{z}, \mathbf{w}' - \mathbf{w} \rangle + \frac{\beta}{2} \|\mathbf{w}' - \mathbf{w}\|, \quad \forall \mathbf{w}' \in S, \forall \mathbf{z} \in \partial f(\mathbf{w})$$

If the function $R(\mathbf{w})$ is differentiable, we can replace $\mathbf{z}$ with $\nabla R(\mathbf{w})$

A convex function could be lower bounded with a hyperplane. A strongly-convex function could be lower bounded with a parabola (quadratic form).

Proposition 1 (Lower bounding a strongly-convex function)**.** *A β-strongly convex function $R : S \to \mathbb{R}$ is defined over domain S, with respect to the norm $\|.\|$. If $\mathbf{w}_{\min} = \arg\min_{\mathbf{w}\in S} R(\mathbf{w})$, then,*

$$R(\mathbf{w}) - R(\mathbf{w}_{\min}) \geq \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}_{\min}\|^2$$

Proof. In the differentiable case, one can replace $\nabla R(\mathbf{w}) = 0$ and get the result. In the nondifferentiable case, we can use the fact that $\mathbf{0} \in \partial f(\mathbf{w}_{\min})$. □

Example 1 (Quadratic function is strongly convex)**.** *It should not be hard to see that a quadratic function is strongly convex. An example is $\frac{1}{2}\|\mathbf{w}\|_2^2$, which is 1-stringly convex.*

Proposition 2. *A sufficient condition is that on being strongly convex is that, if the function is twice differentiable, and its second derivative is more than the second derivative of second order function which can lower bound the it, it is indeed a β-strongly convex function. In other words, if we have*

$$\mathbf{x}^\top \nabla^2 R(\mathbf{w})\mathbf{x} \geq \beta\|\mathbf{x}\|^2$$

for some $\beta > 0$, the function $R(\mathbf{w})$ is strongly convex, with respect to the norm $\|.\|$.

Proof. Using the second-order Taylor approximation and the Definition 2. □

Example 2 (Entropic regulaization is strongly convex)**.** *The regulaizer with the following definition we call entropic regularization:*

$$R(\mathbf{w}) = \sum_{i=1}^{d} w[i] \log w[i], \forall \mathbf{w}, \forall \mathbf{x}$$

where $w[i]$ is the i-th element of the vector $\mathbf{w}$. This function defined over the set $S = \{\mathbf{w} \in \mathbb{R}^d \big| \, \mathbf{w} > 0, \|\mathbf{w}\|_1 \leq \frac{1}{\beta}\}$ for some $\beta > 0$, is β-strongly convex with respect to 1-norm. The proof is simply by using the Proposition 2.

$$\begin{aligned}
\mathbf{x}^\top \nabla^2 R(\mathbf{w})\mathbf{x} &= \sum_{i=1}^{d} \frac{x[i]^2}{w[i]} = \frac{1}{\|\mathbf{w}\|_1}\left(\sum_{i=1}^{d} w[i]\right)\left(\sum_{i=1}^{d} \frac{x[i]^2}{w[i]}\right) \\
&\geq \frac{1}{\|\mathbf{w}\|_1}\left(\sum_{i=1}^{d} x[i]\right)^2 = \frac{\|\mathbf{x}\|_1^2}{\|\mathbf{w}\|_1} \geq \beta\|\mathbf{x}\|_1^2
\end{aligned}$$

which proves the β-strong convexity. The last inequality is using Cauchy-Schwartz inequality.

2.5 Bounding the regret

This section starts the major analysis of the convergence properties for different algorithms.

2.5.1 Bounding FTL

Consider the FTL algorithm in Section 2.1. We start with a lemma. Here for simplicity we denote $l\left(f(x_t;\mathbf{w}_t),z_t\right)\triangleq l_t\left(\mathbf{w}_t\right)$.

Lemma 3. *Suppose running the online algorithm in Section 2.1 would produce the output* $\mathbf{w}'_1,\mathbf{w}'_2,\ldots$*. Then for any* $\mathbf{w}\in S$*:*

$$\begin{aligned}
\mathit{Regret}_T(\mathbf{w}) &= \sum_{t=1}^{T}\left[l_t\left(\mathbf{w}'_t\right)-l_t\left(\mathbf{w}\right)\right]\\
&\leq \sum_{t=1}^{T}\left[l_t\left(\mathbf{w}'_t\right)-l_t\left(\mathbf{w}'_{t+1}\right)\right]
\end{aligned}$$

Proof. We can simplify the objective to the following:

$$\sum_{t=1}^{T} l_t\left(\mathbf{w}'_{t+1}\right)\leq\sum_{t=1}^{T} l_t\left(\mathbf{w}\right),\quad\forall\mathbf{w}\in S \tag{7}$$

We continue proving the lemma using induction on t. The base case is when $T=1$:

$$l_t\left(\mathbf{w}'_2\right)\leq l_t\left(\mathbf{w}\right),\quad\forall\mathbf{w}\in S$$

Since $\mathbf{w}_2=\arg\min_{\mathbf{w}\in S}\sum_{t=1}^{T=1}l_t\left(\mathbf{w}\right)=\arg\min_{\mathbf{w}\in S}l_t\left(\mathbf{w}\right)$, then obviously the base case holds. Suppose the Equation 7 holds for $T=k$:

$$\sum_{t=1}^{k} l_t\left(\mathbf{w}'_{t+1}\right)\leq\sum_{t=1}^{k} l_t\left(\mathbf{w}\right),\quad\forall\mathbf{w}\in S$$

Now we prove that the Equation 7 holds for $T=k+1$: We add $l_t\left(\mathbf{w}'_{k+1}\right)$ to both sides:

$$\sum_{t=1}^{k+1} l_t\left(\mathbf{w}'_{t+1}\right)\leq l_t\left(\mathbf{w}'_{k+1}\right)+\sum_{t=1}^{k} l_t\left(\mathbf{w}\right),\quad\forall\mathbf{w}\in S$$

Since this holds for any $\mathbf{w}$, this holds for the special value of $\mathbf{w}=\mathbf{w}'_{k+2}$, which simplifies the above inequality:

$$\sum_{t=1}^{k+1} l_t\left(\mathbf{w}'_{t+1}\right)\leq\sum_{t=1}^{k+1} l_t\left(\mathbf{w}'_{k+2}\right),\quad\forall\mathbf{w}\in S \tag{8}$$

Also since

$$\mathbf{w}'_{k+2}=\arg\min_{\mathbf{w}\in S}\sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}\right)\Rightarrow\sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}'_{k+2}\right)=\arg\min_{\mathbf{w}\in S}\sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}\right)$$

$$\Rightarrow\sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}'_{k+2}\right)\leq\sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}\right)$$

Combining the Equation 8 with the above, gives us the following:

$$\sum_{t=1}^{k+1} l_t\left(\mathbf{w}'_{t+1}\right) \leq \sum_{t=1}^{T=k+1} l_t\left(\mathbf{w}\right)$$

Which proves the desired result. □

This inequality could be very useful in deriving bounds for algorithms. As an example, consider the following instance, where the loss function is a quadratic function.

Example 3 (FTL with quadratic loss)**.** *Suppose the loss function is defined as the following:*

$$l(\mathbf{z};\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{z}-\mathbf{w}\|_2^2$$

To create a correspondence between what had by now, given a set of observations of $\mathbf{z}_{t-1}$*, we want to choose a* $\mathbf{w}_t$ *which minimizes the regret in the hind side. In less complicated language, we are minimizing this function in online fashion. Suppose there is no constraint. Suppose we have observed* $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$*, and we want to find the FTL prediction for* $\mathbf{w}_k$*, i.e.*

$$\mathbf{w}_k = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{t=1}^{k-1} \frac{1}{2}\|\mathbf{z}_t-\mathbf{w}\|_2^2$$

With simple derivative, we can see that:

$$\mathbf{w}_k = \frac{1}{k-1}\sum_{t=1}^{k-1}\mathbf{z}_t \tag{9}$$

Based on the Lemma 3, we know that if we somehow bound $l(\mathbf{z}_t;\mathbf{w}_k) - l_t(\mathbf{z}_k;\mathbf{w}_{k+1})$*, we can bound the regret for FTL. Let's write the definitions and simplify them:*

$$l(\mathbf{z}_k;\mathbf{w}_k) - l_t(\mathbf{z}_k;\mathbf{w}_{k+1}) = \frac{1}{2}\|\mathbf{z}_k-\mathbf{w}_k\|_2^2 - \frac{1}{2}\|\mathbf{z}_k-\mathbf{w}_{k+1}\|_2^2$$

To simplify this, we use the the formula for $\mathbf{w}_{k+1}$ *:*

$$\begin{cases} \mathbf{w}_k = \frac{1}{k-1}\sum_{t=1}^{k-1}\mathbf{z}_t & \times(k-1) \\ \mathbf{w}_{k+1} = \frac{1}{k}\sum_{t=1}^{k}\mathbf{z}_t & \times k \end{cases} \Rightarrow \begin{cases} (k-1)\mathbf{w}_k = \sum_{t=1}^{k-1}\mathbf{z}_t \\ k\times\mathbf{w}_{k+1} = \mathbf{z}_k + \sum_{t=1}^{k-1}\mathbf{z}_t \end{cases}$$

$$\Rightarrow k\times\mathbf{w}_{k+1} - (k-1)\mathbf{w}_k = \mathbf{z}_k \Rightarrow \mathbf{w}_{k+1} - \mathbf{z}_k = \frac{k-1}{k}\left(\mathbf{w}_k - \mathbf{z}_k\right)$$

$$\Rightarrow \|\mathbf{w}_{k+1}-\mathbf{z}_k\|_2^2 = \left(\frac{k-1}{k}\right)^2 \|\mathbf{w}_k-\mathbf{z}_k\|_2^2$$

Given this, we can simplify the upper bound on the regret:

$$\begin{aligned} l(\mathbf{z}_k;\mathbf{w}_k) - l_t(\mathbf{z}_k;\mathbf{w}_{k+1}) &= \frac{1}{2}\|\mathbf{z}_k-\mathbf{w}_k\|_2^2 - \frac{1}{2}\|\mathbf{z}_k-\mathbf{w}_{k+1}\|_2^2 \\ &= \frac{1}{2}\left[1-\left(\frac{k-1}{k}\right)^2\right]\|\mathbf{w}_k-\mathbf{z}_k\|_2^2 \end{aligned}$$

Let's simplify the coefficient:

$$\frac{1}{2}\left[1-\left(\frac{k-1}{k}\right)^2\right]=\frac{1}{2}\frac{2k-1}{k^2}\leq\frac{1}{k}$$

Since we are trying to find an upper bound on the regret, making this coefficient a little bigger by simplifying it, will not make a problem:

$$l(\mathbf{z}_k;\mathbf{w}_k)-l_t(\mathbf{z}_k;\mathbf{w}_{k+1})\leq\frac{1}{k}\|\mathbf{w}_k-\mathbf{z}_k\|_2^2$$

which gives the following upper bound on the regret:

$$Regret_T\leq\sum_{t=1}^{T}l(\mathbf{z}_t;\mathbf{w}_t)-l_t(\mathbf{z}_t;\mathbf{w}_{t+1})\leq\sum_{t=1}^{T}\frac{1}{t}\|\mathbf{w}_t-\mathbf{z}_t\|_2^2$$

Now suppose we limit out space of options and we have the constraint that $\|\mathbf{w}_t-\mathbf{z}_t\|\leq M$. This constraint comes up when each $\|\mathbf{z}_t\|\leq M/2$, and $\mathbf{w}_t$ average of $\{\mathbf{z}_1,\mathbf{z}_2,\ldots\}$. In this case we will have the following simplified form on the regret:

$$Regret_T\leq M^2\sum_{t=1}^{T}\frac{1}{t}\in O(M^2\times\log T)$$

One important observation is that, the bound does not decrease, as T increases, which is disappointing!

2.5.2 Bounding the FTRL

With a technique similar to the one in the previous subsection, we want to bound the regret for the FTRL introduced in the Section 2.2. For that, consider the following lemma.

Lemma 4. *Suppose running the online algorithm in Section 2.2 would produce the output $\mathbf{w}'_1,\mathbf{w}'_2,\ldots$. Then for any $\mathbf{w}\in S$:*

$$\sum_{t=1}^{T}\left[l_t\left(\mathbf{w}'_t\right)-l_t\left(\mathbf{w}\right)\right]\leq R(\mathbf{w})-R(\mathbf{w}'_1)+\sum_{t=1}^{T}\left[l_t\left(\mathbf{w}'_t\right)-l_t\left(\mathbf{w}'_{t+1}\right)\right]$$

Proof. For proving this we can use the Lemma 3. Running FTRL on $l_1(.),l_2(.),l_3(.),\ldots$ is equivalent to running FTL on $l_0(.)=R,l_1(.),l_2(.),l_3(.),\ldots$. In other words we can just use our upper bound from the Lemma 3, with additional time step $t=0$, which gives the desired bound. □

Example 4 (Regularized linear optimization)**.** *Suppose we are solving the problem a linear loss function $l(\mathbf{z}_t;\mathbf{w})=\langle\mathbf{z}_t,\mathbf{w}\rangle$, where $\langle,\rangle$ is the inner product between its two arguments, and with an additional quadratic regularizer $R(\mathbf{w})=\frac{1}{\beta}\|\mathbf{w}\|_2^2$:*

$$\mathbf{w}_k=\arg\min_{\mathbf{w}\in\mathbb{R}^d}\left[\frac{1}{\beta}\|\mathbf{w}\|_2^2+\sum_{t=1}^{k-1}\langle\mathbf{z}_t,\mathbf{w}\rangle\right]$$

Taking gradient to find the closed form answer, which is $\mathbf{w}_k = \frac{\beta}{2}\sum_{t=1}^{k-1}\mathbf{z}_t$. *Given this choice at each step, can we bound the regret?*
We use the Lemma 4 to bound this FTRL schema. The upper bound on the regret is the following :

$$\begin{aligned} Regret_T &\leq R(\mathbf{w}) - R(\mathbf{w}'_1) + \sum_{t=1}^{T}\left[l_t\left(\mathbf{w}'_t\right) - l_t\left(\mathbf{w}'_{t+1}\right)\right] \\ &= \frac{1}{\beta}\|\mathbf{w}\|_2^2 - 0 + \sum_{t=1}^{T}\langle \mathbf{z}_t, \mathbf{w}_t - \mathbf{w}_{t+1}\rangle \end{aligned}$$

Given our strategy for choosing $\mathbf{w}_k$ *let's simplify* $\mathbf{w}_t - \mathbf{w}_{t+1}$

$$\begin{cases} \mathbf{w}_k = \frac{\beta}{2}\sum_{t=1}^{k-1}\mathbf{z}_t \\ \mathbf{w}_{k+1} = \frac{\beta}{2}\sum_{t=1}^{k}\mathbf{z}_t \end{cases} \quad \Rightarrow \mathbf{w}_{k+1} - \mathbf{w}_k = \frac{\beta}{2}\mathbf{z}_k$$

$$Regret_T \leq \frac{1}{\beta}\|\mathbf{w}\|_2^2 + \frac{\beta}{2}\sum_{t=1}^{T}\|\mathbf{z}_k\|_2^2$$

Suppose we put further limit on $\mathbf{w}$. *In particular we set* $\|\mathbf{w}\| \leq B$ *and* $\frac{1}{T}\sum_{t=1}^{T}\|\mathbf{z}_k\|_2^2 \leq C$.

$$Regret_T \leq \frac{1}{\beta}B^2 + \frac{\beta}{2}TC$$

Now the key point is that one can minimize β *in a way that, makes this function as slow as possible, as a function of the desire variable.*

- *To make the bound grow linearly as function of* B, *add linear term of* B *to* β.
- *To make the bound grow with squared root of* T, *add* $1/\sqrt{T}$ *to* β.
- *To make the bound grow with squared root of* C, *add* $1/\sqrt{C}$ *to* β.

The results is $\beta = \frac{B}{\sqrt{TC}}$. *The resulting upper bound is*

$$Regret_T \leq O(B\sqrt{TC})$$

Example 5 (Regularized quadratic optimization)**.** *Suppose we are solving the problem the same loss as in the Example 3, but with an additional quadratic regularizer* $R(\mathbf{w}) = \frac{1}{\beta}\|\mathbf{w}\|_2^2$

$$\mathbf{w}_k = \arg\min_{\mathbf{w}\in\mathbb{R}^d}\left[\frac{1}{\beta}\|\mathbf{w}\|_2^2 + \sum_{t=1}^{k-1}\frac{1}{2}\|\mathbf{z}_t - \mathbf{w}\|_2^2\right]$$

[DISCONTINUED: removed or add more details]

2.5.3 Linear loss function

Consider the linear loss function represented in Section 2.3. We can prove the following lemma for the generalization bound of this form:

Lemma 5. *Suppose $\mathcal{W} = \mathbb{R}^n$, i.e. unconstrained case. Then for $\mathbf{w} \in \mathcal{W}$:*

$$\frac{1}{\beta} \sum_{t=1}^{T} \mathbf{g}_t^\top (\mathbf{w}_t - \mathbf{w}) \leq D_R(\mathbf{w}, \bar{\mathbf{w}}_1) - D_R(\mathbf{w}, \bar{\mathbf{w}}_{T+1}) + \frac{1}{\beta} \sum_{t=1}^{T} \mathbf{g}_t^\top (\mathbf{w}_t - \mathbf{w}_{t+1})$$

Given this, the prediction for FTRL could be found using the following optimization:

Proof. [PROOF?] □

Proposition 3. *Given a strongly convex regulaizer R with respect to a specific norm $\|.\|$, then for FTRL we have:*

$$\sum_{t=1}^{T} \mathbf{g}_t^\top \left(\mathbf{w}_t - \mathbf{w} \right) \leq \frac{1}{\beta} \sum_{i=1}^{T} \left(\|\mathbf{g}_t\|^* \right)^2 + \beta \left(R(\mathbf{w}) - R(\mathbf{w}_1) \right)$$

If ?? and $\beta = 1/\sqrt{R(\mathbf{w})/T}$:

$$\sum_{t=1}^{T} \mathbf{g}_t^\top \left(\mathbf{w}_t - \mathbf{w} \right) \leq \sqrt{T R(\mathbf{w})}$$

Proof. By definition of strong convexity with respect to the norm $\|.\|$ we have:

$$R(\mathbf{w}_t) \geq R(\mathbf{w}_{t+1}) + \langle \nabla R(\mathbf{w}_{t+1}), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2$$

With the convexity we can lower bound the $R(\mathbf{w})$ function with a hyperplane:

$$R(\mathbf{w}_{t+1}) \leq R(\mathbf{w}_t) + \langle \nabla R(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle$$

Combining the above two inequalities we have:

$$\frac{1}{2} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2 \leq \langle \nabla R(\mathbf{w}_t) - \nabla R(\mathbf{w}_{t+1}), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle$$

[PROOF INCOMPLETE]

□

2.6 Mirror Descent

Proposition 4 (A useful form)**.** *Remember the*

[UP TO HERE] Let's start with some results in concentration inequalities.

3 Online learning for Classification problems

4 Bibliographical notes