CSE599s, Spring 2014, Online Learning Lecture 3 - 04/08/2014

Follow-The-Regularized-Leader

Lecturer: Brendan McMahan *Scribe: Alexandre Bykov*

1 Review of Follow-The-Leader

Last time we analyzed the Follow-The-Leader (FTL) algorithm. As a reminder, below is the rule the player uses at round $t+1$ for picking w_{t+1}:

$$w_{t+1} = \operatorname*{argmin}_{w \in W} f_{1:t}(w).$$

We proved that FTL has the following regret bound:

Lemma 1 (FTL Regret Bound)**.** *Let $w_1, ..., w_T$ be the points played by the player during FTL. Then, $\forall u \in W$*

$$\text{Regret} \leq \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}).$$

There are two key problems with FTL. First, solving for the w that minimizes $f_{1:t}(w)$ could be a hard optimization problem. The bigger issue is that, in general, the regret bound is $O(T)$. This is especially true for linear functions. The poor regret bound comes from the fact that $f_t(w_t) - f_t(w_{t+1})$ will be large if w_{t+1} is far from w_t. This issue can be mitigated by adding regularization to improve the stability of the solution.

2 Follow-The-Regularized-Leader

The Follow-The-Regularized-Leader (FTRL) algorithm is the FTL algorithm with a regularizer term added. Let $R : W \to \mathbb{R}$. At round $t+1$ the player will pick w_{t+1} according to

$$w_{t+1} = \operatorname*{argmin}_{w \in W}(f_{1:t}(w) + R(w)). \tag{1}$$

Without loss of generality we can say that $R(0) = 0$, which minimizes the regularization function. Any regularization function can be translated so that it is minimized at 0. From this fact it follows that

$$w_1 = \operatorname*{argmin}_{w \in W} R(w) = 0. \tag{2}$$

Claim 2. *By inserting the regularization term we can make the solution more stable.*

This claim will be proven later but first we will consider a simple example.

2.1 Example: Linear Loss FTRL

Consider a linear loss function of the form

$$f_t(w) = g_t \cdot w$$

with $g_t \in \mathbb{R}^n$. Let the regularization term be of the form

$$R(w) = \frac{1}{2\eta}\|w\|_2^2$$

where $\eta \geq 0$. For this example we can let $W = \mathbb{R}^n$ since the regularization term will ensure a nice solution. Substituting the above definitions into (1) we get

$$w_{t+1} = \underset{w \in \mathbb{R}^n}{\operatorname{argmin}}(g_{1:t} \cdot w + \frac{1}{2\eta}\|w\|_2^2).$$

To minimize the inside expression we can take the gradient with respect to w and set it to 0:

$$\nabla_w(g_{1:t} \cdot w + \frac{1}{2\eta}\|w\|_2^2) = g_{1:t} + \frac{1}{\eta}w = 0 \Rightarrow w = -\eta g_{1:t}.$$

Therefore we have that at step $t+1$ the player must pick

$$w_{t+1} = -\eta g_{1:t}.$$

Instead of recalculating this quantity at each time step, we can reformulate this in terms of w_t. From the equation above we know that

$$w_t = -\eta g_{1:t-1} \Rightarrow g_{1:t-1} = \frac{-w_t}{\eta}.$$

We can therefore rewrite w_{t+1} as

$$w_{t+1} = -\eta g_{1:t} = -\eta(\frac{-w_t}{\eta} + g_t) = w_t - \eta g_t.$$

This is exactly the same as the formula for gradient descent with a constant learning rate. To check for the stability of these solution vectors we can compute

$$f_t(w_t) - f_t(w_{t+1}) = g_t \cdot (w_t - w_{t+1}) = g_t \cdot (-\eta g_{1:t-1} + \eta g_{1:t}) = g_t \cdot (\eta g_t) = \eta\|g_t\|_2^2. \tag{3}$$

By correctly picking a value for η we can ensure that this value is low.

Observation 3. *For certain loss functions, FTL will also acheive the above regret bound. For example, loss functions of the form*

$$f_t(w) = \frac{1}{2}\|w - z_t\|_2^2$$

where z_t is picked by the adversary, will also create a stable solution. Our quadratic regularizer produces the same effect as a quadratic loss function.

2.2 FTRL Regret Bound

Lemma 4 (FTRL Regret Bound)**.** *Let $w_1, ..., w_T$ be the points played by the player during FTRL. Then, $\forall u \in W$*

$$\text{Regret}(u) \leq R(u) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}).$$

Proof. Play FTL with the following sequence of loss functions: $f_0 = R, f_1, f_2, ..., f_T$. By the way we defined w_t in (1), we can guarantee that FTL will play the exact same sequence $w_1, w_2, ..., w_T$ as FTRL in this situation. From Lemma 1 we know that

$$\sum_{t=0}^{T} f_t(w_t) - f_t(u) \leq \sum_{t=0}^{T} f_t(w_t) - f_t(w_{t+1}).$$

Substituing $f_0 = R$ we get:

$$R(w_0) - R(u) + \sum_{t=1}^{T} f_t(w_t) - f_t(u) \leq R(w_0) - R(w_1) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}).$$

By (2) we know that $R(w_1) = 0$. Substituting that in and simplifying we get:

$$\sum_{t=1}^{T} f_t(w_t) - f_t(u) \leq R(u) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}) \Rightarrow$$

$$\text{Regret}(u) \leq R(u) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}).$$

□

2.3 Linear Loss Function Analysis

Linear loss functions caused some of the worst behavior for FTL. Here we analyze the performance of FTRL on linear loss functions.

Corollary 5. *Consider FTRL with regularizer*

$$R(w) = \frac{1}{2\eta}\|w\|_2^2.$$

Then $\forall u \in \mathbb{R}^n$*:*

$$\text{Regret}(u) \leq \frac{1}{2\eta}\|u\|_2^2 + \sum_{t=1}^{T} \eta\|g_t\|_2^2.$$

Proof. As shown in Lemma 4, we know that

$$\text{Regret}(u) \leq R(u) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1}).$$

By (3) we know that

$$f_t(w_t) - f_t(w_{t+1}) = \eta\|g_t\|_2^2$$

for the specified R. Substituting that and the equation for R into the regret bound completes the proof. □

An important question is whether this regret is sublinear. Clearly this depends on the learning rate that we choose. If the learning rate is too high then the adversary can pick points such that we constantly oscillate around the optimal solution. If the learning rate is too small the adversary can pick an optimal value far away from the initial value and we will never get there. To avoid these issues we will need to pick an optimal step size.

Assumption 6. $\forall t, \|g_t\|_2 \leq G$ *and* $W = \{u | \|u\|_2 \leq B\}$

By the above assumptions and Corollary 5 we get:

$$\text{Regret}(u) \leq \frac{1}{2\eta}\|u\|_2^2 + \sum_{t=1}^{T} \eta\|g_t\|_2^2 \leq \frac{B^2}{2\eta} + \eta T G^2.$$

We can solve for the η that minimizes this regret bound. Taking the derivate of the above expression in terms of η and setting it equal to 0 we get:

$$\frac{d}{d\eta}(\frac{B^2}{2\eta} + \eta T G^2) = -\frac{B^2}{2\eta^2} + TG^2 = 0 \Rightarrow \frac{B^2}{2\eta^2} = TG^2 \Rightarrow \eta = \frac{B}{G\sqrt{2T}}.$$

Substituting this value for η back into the regret bound we get:

$$\text{Regret}(u) \leq \frac{GB^2\sqrt{2T}}{2B} + \frac{BTG^2}{G\sqrt{2T}} = \frac{BG\sqrt{T}}{\sqrt{2}} + \frac{BG\sqrt{T}}{\sqrt{2}} = BG\sqrt{2T}.$$

Therefore the regret is sublinear and is in fact $O(\sqrt{T})$. This analysis highlights that picking a correct learning reate is crucial for online gradient descent. Estimating the constants D, G and T can be difficult in practice and there has been a lot of work to solve that issue. By picking an η that varies with t we can achieve a regret that is almost as good without knowing T in advance. Furthermore, T and G can be eliminated by rephrasing the problem in terms of g_t. These topics will be covered in future lectures.

3 Generalizing Beyond Linear Loss Functions

We can envision the online learning task as the following diagram:

Figure 1: Block diagram for online learning game

The arrows illustrate one round of the online learning game. First, the adversary sends a loss function f_t. Next it gets transformed in the transformation block and the transformed $\hat{f}_t$ is sent to the online algorithm. The online algorithm responds with a point $\hat{w}_{t+1}$. This point may then have to be transformed again into the w_{t+1} that is sent back to the adversary. The dashed box represents the parts that we have control over. Through our previous analysis we have shown that the FTRL algorithm can be envisioned as a transformation of the FTL algorithm. Below is the diagram for FTRL:

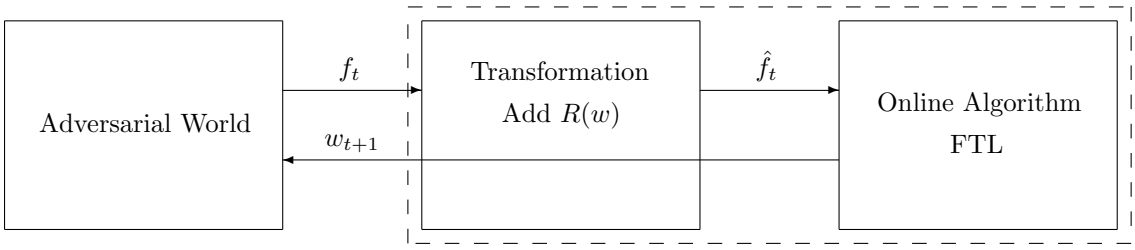

Figure 2: Block diagram for FTRL

The trasnformation reflects adding a regularizer term to the standard FTL algorithm. As can be seen in the diagram, w_{t+1} does not need to be transformed on the way back to the adversary.
Similar to the FTL to FTRL transformation, we can think of generalizing the FTRL algorithm by transforming convex loss functions into linear loss functions that we already know how to analyze.

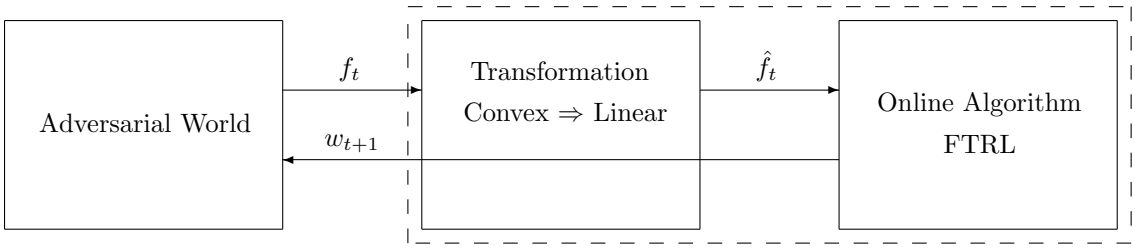

Figure 3: Block diagram for FTRL with convex loss functions

As in Figure 2, we don't need to transform w_{t+1} on the way back to the adversary. To determine the correct transformation we first need to review convexity.

4 Convexity

Definition 7. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\forall \alpha \in [0,1]$ and $\forall u, w \in \mathbb{R}^n$:*

$$f(\alpha w + (1-\alpha)u) \leq \alpha f(w) + (1-\alpha) f(u).$$

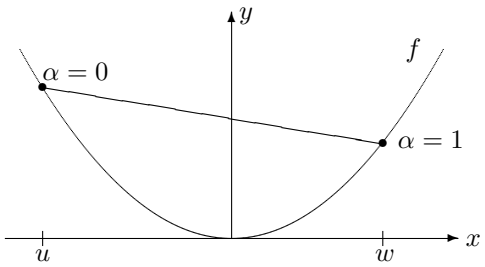

Figure 4: Illustration of convexity definition

The curve represents a convex function f and the line above it is $y = \alpha f(w) + (1-\alpha) f(u)$ for varying values of α. The above definition states that a convex function must be below that line for any choices of u and w. A key property of convex functions (that will turn out to be very useful for transforming them into linear functions) is that a tangent line drawn at any point will always be below the function and will approximate it well in a small neighborhood around the tangent point. Differentiability is not necessary for this property to hold. For example, the absolute value function is not differentiable at the origin, however any tangent line through the origin satisfies this property.

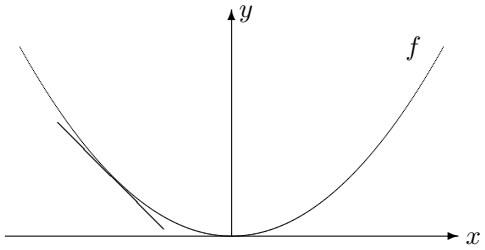

Figure 5: Illustration of tagent line property