CSE599s, Spring 2012, Online Learning Lecture 8 - 04/19/2012

The FTRL Algorithm with Strongly Convex Regularizers

Lecturer: Brandan McMahan *Scribe: Tamara Bonaci*

1 Introduction

In the last lecture, we talked about regularization models that induce sparsity, and we explained why such models might be preferred:

- *In statistics,* a sparse vector might correspond to a feature selection. For example, when there are more features than training examples, we might choose to set some features to 0 using L_1-regularization.
- *From systems perspective*, having a large number of features requires storing a large number of coefficients. By setting some features to 0, we may reduce memory requirements.

We also talked about the difference between L_1- and L_2- regularization, and presented a version of an Online Gradient Descent (OGD) algorithm, that includes a prediction error and an L_1- regularization term. For the most part today, our **loss function** will be defined as

$$f_t(\omega) = \underbrace{\ell_t(\omega)}_{\text{prediction error}} + \underbrace{\lambda\|\omega\|_1}_{\text{regularization term}}, \tag{1}$$

where $\ell_t(\omega)$ represents a short-hand notation for $\ell_t(\omega) := \ell(\omega \cdot x_t, y_t)$, $\lambda\|\omega\|_1$ is a regularization term used to induce sparsity and $\lambda \in \mathbb{R}$ is a weighting factor.

In some sense, this loss function allows us to choose predictors such to minimize our regret compared to the best model making a prediction, while ensuring our predicted vector is sparse. (For example, if there exists a perfect predictor ω^* that makes a good prediction every time, but has a large L_1- norm, we might not care about our regret compared to that predictor as much.)

Remark 1: The weighting factor, λ, is highly dependant on the problem we are solving and, for now, we won't care about it. As a general rule, however, we note that a hypothetical sparsity/loss curve parameterized by λ has a form depicted in Figure 1.

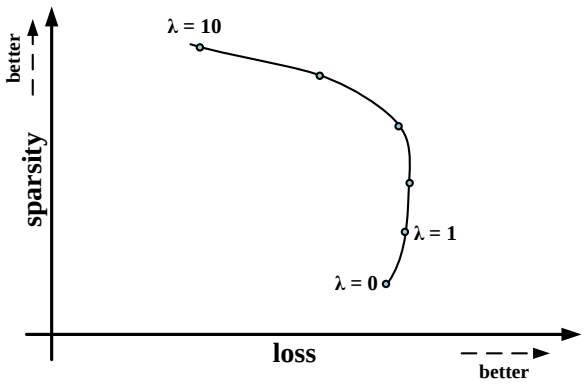

Figure 1: A sparsity/loss curve parameterized by the weighting factor λ.

2 Composite Objective-OGD Algorithm

Let's consider the OGD algorithm again, with the update rule given as

$$\omega_{t+1} = \omega_t - \eta \hat{g}_t, \tag{2}$$

where $\hat{g}_t$ denotes a subgradient of the loss function $f_t(\omega)$, $\hat{g}_t \in \partial f_t(\omega_t)$. We've analyzed this algorithm last time and observed it can results in oscillatory predictions, as there is nothing to really force the predictor to got exactly to 0. In fact, if there is any noise in the loss function gradient, the predictor will never go to 0.

We can, however, rewrite the ODG update rule (2) as the following optimization problem

$$\omega_{t+1} = \underset{\omega}{\operatorname{argmin}}\, g_t\omega + \frac{1}{2\eta}\|\omega - \omega_t\|^2, \tag{3}$$

where we approximate the prediction error, $f_t(\omega)$, by its subgradient at ω_t, but keep the sparsity-regularization part intact, $g_t\omega + \lambda\|\omega\|_1$, and $g_t \in \partial f_t(\omega_t)$. This modified algorithm is known as the **Composite-Objective Online Gradient Descent (CO-OGD)**.

The CO-OGD algorithm (3) helps prevent prediction oscillations. It turns out, however, that we can achieve even sparser predictor vector using the Follow-the-Regularized-Leader (FTRL) algorithm, with the update rule defined as [3]

$$\omega_{t+1} = \underset{\omega}{\operatorname{argmin}}\Big(\underbrace{g_{1:t}\omega}_{\text{subgradient approximation}} + \underbrace{t\lambda\|\omega\|_1}_{t \text{ copies of } L_1 \text{ penalty}} + \underbrace{r_{1:t}(\omega)}_{\text{stabilization penalty}} \Big). \tag{4}$$

In order to compare the CO-OGD with the FTRL algorithm and show that the FTRL algorithm (4) enjoys a better sparsity, we next rewrite the CO-OGD algorithm (3) in the following *alternative form*:

$$\omega_{t+1} = \underset{\omega}{\operatorname{argmin}}(g_{1:t}\omega + \phi_{1:t-1}\omega + \lambda\|\omega\|_1 + r_{1:t}(\omega)), \tag{5}$$

with $r_t(\omega)$ defined as $r_t(\omega) = \frac{\sigma_t}{2}\|\omega - \omega_t\|^2$ and $\phi_t \in \mathbb{R}^n$ as a subgradient approximation of the L_1- penalty, $\phi_t \in \partial(\lambda\|\omega\|_1)$. Thus, $\phi_{1:t-1}\omega + \lambda\|\omega\|_1$ approximates the term $t\lambda\|\omega\|_1$ from the FTRL algorithm (4).

While the fact that CO-OGD update rule (3) and its alternative form (5) are the same is not immediately obvious (and we do not prove it here), using the alternative representation allows us to immediately see that in the CO-OGD there is a single L_1- penalty for the current learning round, and all previous rounds are approximated using subgradient (linear) approximation. On the other hand, in the FTRL algorithm there are t copies of the L_1- penalty, for each iteration of the game. Thus, the FTRL algorithm results in a sparser solution, since it does not assume approximation of the L_1- regularization on any of the iterations of the game.

Remark 2: Note that it is possible to implement this modified FTRL algorithm by storing only a single vector in $\mathbb{R}^n$, or two vectors in $\mathbb{R}^n$, if using adaptive per-coordinate learning rate. Thus, both FTRL and CO-OGD have the same storage requirements.

Remark 3: [Geometrical interpretation of the stabilization penalty] After t rounds, the cumulative stabilization penalty is given as

$$r_{1:t}(\omega) = \sum_{s=1}^{t} \frac{\sigma_s}{2}\|\omega - \omega_s\|^2.$$

This stabilization penalty corresponds to a slowly decreasing learning rate, which can geometrically be represented as depicted in Figure 2, where ω_i denotes the predictor value in the current round.

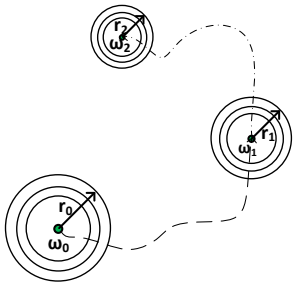

Figure 2: Geometrical interpretation of the stabilization penalty.

3 The FTRL Algorithm with Strongly Convex Proximal Regularizers

We next analyze the FTRL algorithm with a strongly convex proximal regularizer, and derive its regret bounds. As a part of our regret analysis, we also show a simple way of defining general convex feasible sets. We start by defining strongly convex functions.

3.1 Strongly Convex Functions

Definition 1. *A convex function, f, is $\sigma-$* **strongly convex** *with respect to some norm, $\|\cdot\|$, over a set $\mathcal{W}$ if for all $u, w \in \mathcal{W}$, and every g such that $g \in \partial f(w)$ it holds that*

$$f(u) \geq f(w) + g \cdot (u - w) + \frac{\sigma}{2}\|u - w\|^2. \tag{6}$$

A graphical interpretation of strong convexity is depicted in Figure 3.

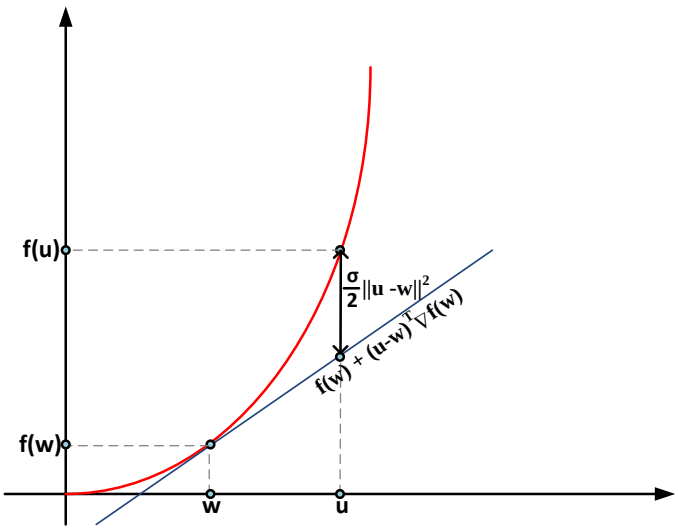

Figure 3: A graphical interpretation of strong convexity.

Remark 4: Strongly convex functions can be thought of as functions that, in addition to a linear lower bound, also have a quadratic lower bound.

Remark 5: We will almost always work with strongly convex regularization functions, but we will not assume the same about the loss functions.

Example: Function $f(w) = \frac{1}{2}\|w\|_2^2$ is 1-strongly convex with respect to Euclidean norm.

Proof. Function $f(w)$ is differentiable, so its subdifferential consists of only one element, $\nabla f(w) = w$. Now, for some points $u, w \in \mathbb{R}$, we can write:

$$\frac{1}{2}u^2 \geq \frac{1}{2}w^2 + w(u-w) + \frac{1}{2}(u-w)^2,$$
$$\frac{1}{2}u^2 \geq \frac{1}{2}u^2.$$

□

Lemma 2. *Let $\mathcal{W}$ be a nonempty convex set, and let $f : \mathcal{W} \to \mathbb{R}$ be a $\sigma-$strongly convex function over $\mathcal{W}$ with respect to a norm $\|\cdot\|$. Further, let w^* be the minimizer of f over $\mathcal{W}$, $w^* = \operatorname{argmin}_{v\in\mathcal{W}} f(v)$. Then for every $u \in \mathcal{W}$ it holds that*

$$f(u) - f(w^*) \geq \frac{\sigma}{2}\|u - w^*\|^2. \tag{7}$$

Proof. (Proof adapted from [2]) Let's first assume f is differentiable and w^* is in the interior of $\mathcal{W}$. Then $\nabla f(w) = 0$ and from the definition of strong convexity it follows that

$$\forall u \in \mathcal{W}, \quad f(u) - f(w^*) \geq \frac{\sigma}{2}\|u - w^*\|^2, \quad \text{as required.}$$

Let's now consider the case when w^* is on the boundary of $\mathcal{W}$. We still have that for all $u \in \mathcal{W}, \nabla f(w^*)^T(u - w^*) \geq 0$, otherwise w^* wouldn't have been optimal since we could make a small step in the direction $u - w^*$ and decrease the value of f. So, the desired inequality still holds.

Finally, let's consider the case when f is not differentiable. Let $g : \mathbb{R}^d \to \mathbb{R}\bigcup\{\infty\}$ be such that $g(w^*) = f(w^*)$ if $w^* \in \mathcal{W}$ and $g(w^*) = \infty$ otherwise. We can therefore rewrite $w = \operatorname{argmin}_v g(v)$. Since g a proper convex function (never receives value $-\infty$), we know that $0 \in \partial g(w^*)$. Thus, the inequality (7) follows using the strong convexity of g. □

3.2 Regret Analysis of the FTRL Algorithm with Strongly Convex Regularizers

We next proceed with the regret analysis of the FTRL algorithm with the following update rule

$$\omega_{t+1} = \operatorname*{argmin}_{\omega\in\mathbb{R}^n} f_{1:t}(\omega) + r_{1:t}(\omega).$$

We further assume the following about loss and regularization functions:

- The loss f_t is L_t- Lipschitz and convex,
- The regularization function is σ_t- strongly convex
- $0 \in \partial r_t(\omega_t)$

Lemma 3 (FTRL lemma)**.** *The regret of the FTRL algorithm with a strongly convex regularizer, compared to some predictor, w^*, satisfies*

$$\textit{Regret}(FTRL) \leq \sum_{t=1}^{T}(f_t(w_t) - f_t(w_{t+1})) + r_{1:T}(w^*). \tag{8}$$

Proof. The proof is analgous to the simpler version of the lemma with a single fixed regularizer r. □

Theorem 4. *Let $f_1, \dots, f_T$ be a sequence of convex functions such that f_t is L_t- Lipschitz with respect to some norm $\|\cdot\|$. Assume that the FTRL algorithm is run on the sequence with a regularization function which is $\sigma-$strongly convex with respect to the same norm. Then, for all $u \in \mathcal{W}$*

$$Regret_T(u) \leq \sum_{t=1}^{T} \frac{L_t^2}{\sigma_{1:t}} + r_{1:T}(\omega^*). \tag{9}$$

Proof. Let's first recall the regret bound for the FTRL algorithm that we derived in Lemma 3

$$\sum_{t=1}^{T}(f_t(\omega_t) - f_t(u)) \leq \sum_{t=1}^{T}(f_t(\omega_t) - f_t(\omega_{t+1})) + r_{1:T}(\omega^*). \tag{10}$$

If the loss function, f_t, is L_t-Lipschitz with respect to a norm $\|\cdot\|$, then:

$$f_t(\omega_t) - f_t(\omega_{t+1}) \leq L_t\|\omega_t - \omega_{t+1}\|. \tag{11}$$

Thus, in order to achieve a small regret, we need to ensure that $\|\omega_t - \omega_{t+1}\|$ is small. If the regularization function is strongly convex with respect to the same norm, that is indeed the case, since ω_t is close to ω_{t+1}. To show that ω_t is close to ω_{t+1}, let's fix t and let's define a *helper function*, h_t, as

$$h_t(\omega) = \underbrace{f_{1:t-1}(\omega)}_{t-1 \text{ losses}} + \underbrace{r_{1:t-1}(\omega) + r_t(\omega)}_{t \text{ regularizations}}. \tag{12}$$

By assumption, $0 \in \partial r_t(\omega_t)$. This implies that h_t is σ_t- strongly convex, and the update rule for ω_t can be rewritten using the helper function, h_t

$$\omega_t = \operatorname*{argmin}_w h_t(\omega).$$

Similarly, let's also rewrite the update rule for ω_{t+1}

$$\omega_{t+1} = \operatorname*{argmin}_\omega h_t(\omega) + f_t(\omega).$$

Since the sum of a convex and a strongly convex function remains strongly convex [2], we know $h_t(\omega_t)$ and $h_t(\omega_{t+1})$ are $\sigma_{1:t}-$ strongly convex. We can thus apply Lemma 2 to h_t (with minimizer ω_t)

$$h_t(\omega_{t+1}) - h_t(\omega_t) \geq \frac{\sigma_{1:t}}{2}\|\omega_t - \omega_{t+1}\|^2. \tag{13}$$

Repeating the same argument for the helper functions h_{t+1}, we can write

$$h_{t+1}(\omega_t) - h_{t+1}(\omega_{t+1}) \geq \frac{\sigma_{1:t}}{2}\|\omega_t - \omega_{t+1}\|^2. \tag{14}$$

Summing inequalities (13) and (14), we obtain

$$f_t(\omega_t) - f_t(\omega_{t+1}) \geq \sigma_{1:t}\|\omega_t - \omega_{t+1}\|^2. \tag{15}$$

Now, using the Lipschitzness of the loss function f_t (inequality (11)), we obtain

$$L_t\|\omega_t - \omega_{t+1}\| \geq f_t(\omega_t) - f_t(\omega_{t+1}). \tag{16}$$

Combining inequalities (15) and (16), we can further write

$$\|\omega_t - \omega_{t+1}\| \leq \frac{L_t}{\sigma_{1:t}}. \tag{17}$$

Now, combining inequalities (16) and (17), we get

$$f_t(\omega_t) - f_t(\omega_{t+1}) \leq \frac{L_t^2}{\sigma_{1:t}}. \tag{18}$$

Combining inequality (18) with the regret bound (10), we get

$$\text{Regret}_T(u) \leq \sum_{t=1}^{T} \frac{L_t^2}{\sigma_{1:t}} + r_{1:T}(\omega^*). \tag{19}$$

Inequality (19) completes the proof. □

Let's now choose regularization function as $r_t(\omega) = \frac{\sigma_t}{2}\|\omega - \omega_t\|^2$. The cumulative sum of regularizers becomes

$$r_{1:T}(\omega^*) = \sum_{t=1}^{T} \frac{\sigma_t}{2}\|\omega - \omega_t\|^2.$$

Let's further assume that for every $\omega \in \mathcal{W} \Rightarrow \|\omega\|_2 \leq R$. It follows that the cumulative sum is upper-bounded by

$$r_{1:T}(\omega^*) = \sum_{t=1}^{T} \frac{\sigma_t}{2}\|\omega - \omega_t\|^2 \leq \sum_{t=1}^{T} \frac{\sigma_t}{2}(2R)^2 = 2\sigma_{1:T}R^2.$$

The regret bound (9) can now be rewritten as

$$\text{Regret} \leq \sum_{t=1}^{T} \frac{L_t^2}{\sigma_{1:t}} + 2\sigma_{1:t}R^2. \tag{20}$$

Taking a learning-rate schedule η_t for gradient descent, we can choose σ_t such that

$$\frac{1}{\sigma_{1:t}} = \eta_t,$$

where η_t denotes the *learning rate*. The regret bound (20) now becomes

$$\text{Regret} \leq \sum_{t=1}^{T} L_t^2\eta_t^2 + \frac{2R^2}{\eta_T}. \tag{21}$$

Note that if $f_t(\omega) = g_t\omega$, then $\|g_t\| = L_t$. Thus, the lower bounds on the regret of the OGD algorithm with adaptive learning rates and the regret of the FTRL algorithm with strongly convex proximal regulizers differ by at most factor of $\frac{1}{2}$.

Remark 6: The scaling parameter σ_t now becomes a part of the regret function. Larger σ_t (smaller learning rate) produces a more stable algorithm, but it may take longer to move across the feasible set, resulting in a larger penalty when ω^* is far away.

3.3 Feasible Sets

We can define proximal regularizers r_t as 1-strongly convex functions such that $0 \in \partial r_t(0)$ (e.g., $r_t(w) = \frac{1}{2}\|w\|^2$). Let's rewrite r_t as

$$r_t(w) = \frac{\sigma_t}{2} r(w - w_t) + I_w(w), \tag{22}$$

where $I_{\mathcal{W}}(x)$ denotes an *indicator function* that ensures a player always plays a point form the feasible set $\mathcal{W}$

$$I_{\mathcal{W}}(w) = \begin{cases} 0, & w \in \mathcal{W}, \\ \infty, & \text{otherwise.} \end{cases}$$

Such an indicator function works for an arbitrary convex sets and does not hurt the regret bound.

Let's now apply this same idea on the FTRL algorithm; given a loss function $f_t(\omega) = \ell_t(\omega) + \lambda\|\omega\|_1$, let's redefine it as

$$\hat{f}_t(\omega) = \ell_t(\omega_t) + g_t(w - w_t) + \lambda\|\omega\|_1,$$

where $g_t(\cdot)$ denotes a subgradient approximation of ℓ_t, $g_t := \nabla\ell_t(\omega_t)$. We can run the FTRL algorithm using $\hat{f}_t(w)$ and get the same regret bound, since:

- $\hat{f}_t(\omega_t) = f_t(\omega_t)$
- $\hat{f}_t(\omega^*) \leq f_t(\omega^*)$

Since constants do not contribute to the optimization, without loss of generality, we can write

$$\hat{f}_t(\omega) = g_t\omega + \lambda\|\omega\|_1. \tag{23}$$

So, we get

$$\hat{f}_{1:t}(\omega) = g_{1:t}(\omega) + t\lambda\|\omega\|_1, \tag{24}$$

and for such a loss function, we get the same update as with the FTRL algorithm.

References

[1] N. Cesa-Bianchi and G. Lugosi, "Prediction, Learning, and Games", *Cambridge University Press*, 2006.

[2] S. Shalev-Shwartz, "Online Learning and Online Convex Optimization", *Foundations and Trends in Machine Learning*, 2012.

[3] H. B. McMahan, "Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L_1 Regularization", *Proceedings of the 14th International Conference on Artificiall Intelligence and Statistics (AISTATS)*, 2011.