CSE599s, Spring 2014, Online Learning Lecture 11 - 05/06/2014

Analyzing Adaptive Algorithms

Lecturer: Brendan McMahan *Scribe: Ajay Menon*

1 Recap of Course So Far

We started with Online Linear Optimization, where the adversary gave us a sequence of linear functions. Our solution for this class of problems was to use FTRL with a quadratic regularizer.

Then we wanted to handle arbitrary convex non-linear functions, since that allows us to analyze a much larger class of loss functions. Our solution to this was Online Convex Optimization. The trick here is that we linearized the loss functions f_t, and then applied FTRL to the linearized loss functions. This is essentially Online Gradient Descent (FTRL with quadratic regularizer running on linearized convex loss functions).

After this, we wanted to deal with Strongly Convex Regularizers, so we covered the setting where the player can interact with the world and receive feedback. The Experts Problem was the setting where after we chose an expert, we received full feedback from all the experts (even the ones we didn't choose). The Bandits Problem was the setting where after we chose an action, we only received feedback for the action we chose (partial feedback). Based on our theory regarding strongly convex regularizers, we used the entropic loss since it was strongly-convex with respect to the 1-norm.

Last time, we extended this analysis to Bandits with Expert Advice, where we added a layer of indirection.

2 Slightly Tighter Analysis on Regret Bounds for OLO / OCO with Linearized Functions

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

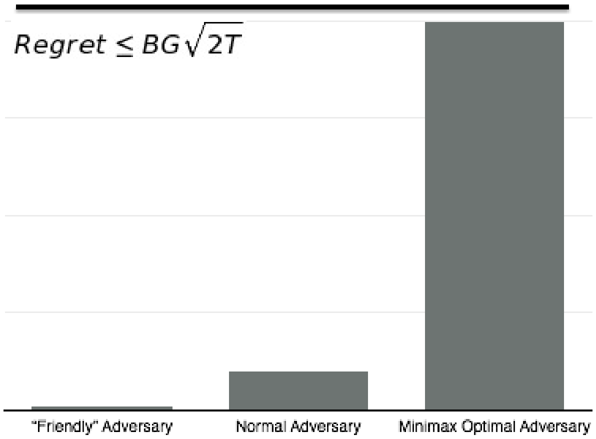
We made some strong assumptions by bounding the norm of the comparator $\|u\|_2 \leq B$, the norm of the gradients $\|g_t\|_2 \leq G$, and the number of rounds we were playing as T. Then we could determine a value for η and simplify our regret bound. We chose $\eta = \frac{B}{G\sqrt{2T}}$. Plugging this value into the original formula produced a new bound of $BG\sqrt{2T}$.

In Homework 1 we used the "Guess and Double" method when trying to find the ideal value for T, and the dependence on T in our value for η prevented us from doing better. If we knew the value of $\eta \sum_{t=1}^{T} \|g_t\|_2^2$, then we could get a better bound on regret.

Another problem we had with the previous regret bound was concatenation. In the homework we tried not only using different w for weekdays and weekends, but also running two completely separate instances of OGD. We found that the regret bounds when using two separate instances was actually slightly better by some constant factors than when we used a single instance with two models. This was because the second model was able to utilize two different learning rates. Now we're going to see how allowing η to change over time can improve our regret bounds.

3 Analyzing Adaptive Algorithms - Tuning η Over Time

The following figure shows how your regret might vary based on the type of adversary. Even though the bound for regret is $O(BG\sqrt{2T})$, in practice you'd hope to have a regret that varies based on how difficult the adversary is.



Realistically you'd expect to be doing much better versus a friendly adversary than a minimax optimal adversary, e.g. the friendly adversary played a lot of gradients with value 0. The ideal situation is to have a data-dependent bound on regret. To get this kind of bound we will need to modify η over time.

Example 1 OGD with a fixed η is relatively bad, and can get linear regret. We set $\eta = \frac{B}{G\sqrt{2T}}$, but we only play $t \leq T$ rounds.

Let's lower bound the regret for OGD.

Choose $f_t(w) = -Gw$ for all t, and $\|g_t\|_2 \leq G$ as a tight bound. Our update rule is $w_{t+1} = t\eta G$, Let the feasible set for the weights be in $[-B, +B]$. The ideal comparator u plays $+B$ at each step.

$$\text{Regret}(t) \leq \sum_{s=1}^{t}((-(-GB) + (-G^2 s\eta)))$$

$$= \sum_{s=1}^{t} G(B - s\eta G) = tGB - \eta G^2 \frac{t(t+1)}{2} = tG(B - \eta G\frac{t+1}{2})$$

$$\geq tG(B - (\frac{B}{G\sqrt{2T}})(\frac{G\sqrt{2T}}{2})) = tG(\frac{B}{2}) = O(t)$$

We can assume that $w_{t+1} = t\eta G \leq B$, so $t + 1 \leq \frac{B}{G}\frac{1}{\eta} = \sqrt{2T}$.

From this example we can see that Online Gradient Descent with a fixed η does well "eventually", but results in poor performance for $t << T$.

Goal: If we knew the number of rounds T and the gradients the adversary played at each round g_t, then we could tighten our regret bound for all T.

$$\text{Regret} \leq B\sqrt{\sum_{t=1}^{T}\|g_t\|_2^2}$$

This regret bound was reached by using the following value of η.

$$\eta = \frac{B}{\sqrt{\sum_{t=1}^{T}(\|g_t\|_2^2)}}$$

In a practical setting we don't know what the final value of T is, but what we can do is let η vary based on our current round T.

$$\eta_T = \frac{B}{\sqrt{\sum_{t=1}^{T}(\|g_t\|_2^2)}}$$

With this adaptive formula η is decreasing by roughly $\frac{1}{\sqrt{T}}$ at each step.

4 Algorithms with Adaptive Learning Rates

We are going to look at a new family of algorithms with adaptive learning rates:

- OGD with adaptive learning rates $\eta_1, \eta_2, \ldots$
  $\bar{w}_{t+1} = \bar{w}_t - \eta_t g_t$, where we pick η_t based on $g_1, \ldots g_t$.
  This is equivalent to: $\bar{w}_{t+1} = \text{argmin}_w(g_t \cdot w + \frac{1}{2\eta_t}\|w - \bar{w}_t\|_2^2)$, with the assumption that $g_t \in \partial f_t(\bar{w}_t)$.
- FTRL-Proximal
  $w_{t+1} = \text{argmin}_w(g_{1:t} \cdot w + \sum_{s=1}^{t}(\frac{\sigma_s}{2}\|w - w_s\|_2^2)$ where $\eta_t = \frac{1}{\sigma_{1:t}}$.
  σ_t = refers to how much change has occurred in the regularizer. e.g. $\sigma_t = 0 \rightarrow \eta_t$ stayed the same.
  FTRL-Proximal is equivalent to OGD when ignoring feasible sets.
- FTRL
  $w_{t+1} = \text{argmin}_w(g_{1:t} \cdot w + \frac{\sigma_{1:t}}{2}\|w\|_2^2)$.
  The regularizer here is $\frac{1}{\eta_t}$ strongly-convex.

4.1 Unified Analysis for OGD and FTRL-Proximal

Lemma 1 (FTRL-Proximal and OGD with Adaptive η_t are Equivalent)**.** *Let w_t be the weight played by FTRL-Proximal on round t. Let $\bar{w}_t$ be the weight played by OGD with η_t learning rate on round t.*

$$w_t = \bar{w}_t.$$

Proof. We can assume without loss of generality that $w_1 = \bar{w}_1$.

Base case: $w_1 = \bar{w}_1 = 0$.

Inductive case:

First define helpful notation:

$$h_{1:t} = g_{1:t} \cdot w + \sum_{s=1}^{t}(\frac{\sigma_s}{2}\|w - w_s\|_2^2)$$

$$h_s(w) = g_s \cdot w + \frac{\sigma_s}{2}\|w - w_s\|_2^2$$

Since the current w on round t is a minimizer for time $t-1$, we know that

$$\nabla h_{1:t-1}(w)) = g_{1:t-1} + \sigma_{1:t-1}w_t - \sum_{s=1}^{t-1}(\sigma_s w_s) = 0$$

because of the inductive hypothesis.

It's sufficient to show that $\nabla h_{1:t}(\bar{w}_{t+1}) = 0$.

From the inductive hypothesis, we know that

$$\bar{w}_{t+1} = g_{1:t} + \sigma_{1:t}(w_t - \frac{1}{\sigma_{1:t}}g_t) - \sum_{s=1}^{t}\sigma_s w_s = w_t - \eta_t g_t = w_t - \frac{1}{\sigma_{1:t}}g_t.$$

Plugging this into $\nabla h_{1:t}(\bar{w}_{t+1})$ we get:

$$\nabla h_{1:t}(\bar{w}_{t+1}) = g_{1:t} + \sigma_{1:t}w_t - g_t - \sum_{s=1}^{t}\sigma_s w_s$$

$$= g_{1:t-1} + \sigma_{1:t-1}w_t + \sigma_t w_t - \sum_{s=1}^{t}\sigma_s w_s$$

$$= g_{1:t-1} + \sigma_{1:t-1}w_t - \sum_{s=1}^{t-1}\sigma_s w_s$$

$$\equiv \nabla h_{1:t-1}(w_t) = 0.$$

□

5 Analysis for General Adaptive FTRL

Given a sequence of functions $r_0, r_1, .., r_T$ (incremental regularizers) such that $r_s : \mathbb{R}^d \to \mathbb{R}$ and $\forall_s r_s(w) \geq 0$.
r_t can be chosen adaptively based on $g_1, \ldots, g_t$.
For FTRL Dual Averaging we can write the incremental regularizer as $r_s(w) = \frac{\sigma_s}{2}\|w\|_2^2$.
For FTRL-Proximal we can write the incremental regularizer as $r_s(w) = \frac{\sigma_s}{2}\|w - w_s\|_2^2$.

Using the notation for incremental regularizers, we can rewrite w_1 as

$$w_1 = \underset{w}{\operatorname{argmin}}(r_0(w))$$

and w_{t+1} as

$$w_{t+1} = \underset{w}{\operatorname{argmin}}(f_{1:t}(w) + r_{0:t}(w)).$$

Note: In this case these f_t are full functions, but they could be linearized into $(g_{1:t} \cdot w)$

2-Part Analysis:
Recover previous results from non-adaptive FTRL:
1) Set $r_0(w) = R(w)$
2) Set $\forall_{t\geq 1} r_t(w) = 0$

5.1 General Adaptive FTRL Analysis - Part 1

First we define some helper functions to simplify the notation when using incremental regularizers.

$$h_{0:t} = f_{1:t}(w) + r_{0:t}(w).$$

$$h_s(w) = f_s(w) + r_s(w).$$

Lemma 2 (Standard FTRL Lemma - See Previous Lectures for Proof)**.**

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

Lemma 3 (Strong FTRL Lemma)**.** *For arbitrary f_t, $r_t(w) \geq 0$*

$$\text{Regret}(u) \leq r_{0:t}(u) + \sum_{t=1}^{T}(h_{0:t}(w_t) - h_{0:t}(w_{t+1}) - r_t(w_t)).$$

Proof. From our definition above, we can see that

$$r_{0:t}(u) = R(w).$$

Expanding the term in the inner summation for the lemma we get:

$$h_{0:t}(w_t) - h_{0:t}(w_{t+1}) - r_t(w_t) = h_{0:t-1}(w_t) - h_{0:t-1}(w_{t+1}) + h_t(w_t) + h_t(w_{t+1}) - r_t(w_t)$$

$$= f_t(w_t) + r_t(w_t) - f_t(w_{t+1}) - r_t(w_{t+1}) + h_{0:t-1}(w_t) - h_{0:t-1}(w_{t+1}) - r_t(w_t).$$

Since w_t is the minimizer for round $t-1$, we know that $h_{0:t-1}(w_t) = 0$. Simplifying this further yields

$$f_t(w_t) - f_t(w_{t+1}) - r_t(w_{t+1}) - h_{0:t-1}(w_{t+1}).$$

By definition $r_t(w_{t+1}) \leq 0$, and $h_{0:t-1}(w_{t+1}) \leq 0$. By taking these simplifications and using them in the original regret bound we get

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

This is exactly the bound we proved in the regular FTRL lemma. □

To be continued...