17-740 Spring 2025
Algorithmic Foundations of Interactive Learning

Lecture 5: Follow the Regularized Leader (cont.)

Lecturer : Steven Wu Scribe: Lintang S., Alfredo G.

5.1 Introduction

- Today: This is the last lecture for online learning covering up to online convex optimization.
- Later: In future lectures we will see how these theoretical tools can be used to derive "idealized" algorithm, serving as a gateway to practical approximations.

Recall from last time the Follow-the-leader algorithm, in which the leader is greedily picking the action and expert that looks the best so far. This however can be *unstable* as the best expert/action may frequently change.

The key idea: Follow-the-regularized-leader (FTRL) is "optimization with *stability*", where the stability is induced via strong convexity.

Revisiting the problem setting, we are making decisions alongside N experts for $1 \leq t \leq T$.

At every round t the learner chooses a probability vector $x^t = (x_1^t, x_2^t, \dots, x_n^t)$. We say that $x^t \in \Delta(N)$, or the probability simplex of experts, representing the set of probability distributions of N experts.

Meanwhile, an adversary reveals l_t incurring some loss $l_t(x^t) =< x^t, l^t >$, where $\forall t, l^t = (l_1^t, l_2^t, \dots, l_n^t) \in [0,1]^N$.

5.1.1 Recap on algorithms

Consider the regret function of choosing expert x from time step $1 \leq t \leq T$.

$$\text{Regret}(x^{1:T}) = \sum_{t=1}^{T} l_t(x^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} l_t(x).$$

As the regret function measures a sum of T difference of losses, we seek to find a learner for which regret grows sub-linearly, and as a result,

$$\frac{Regret(x^{1:T})}{T} \to 0.$$

As we observed last time, there are a handful of algorithms for which we may achieve a diminishing regret.

1. Multiplicative weights (update method): In this first approach the probability distributions for the experts is chosen proportional to the losses incurred in previous time steps:

$$x_i^t \propto \prod_{\tau=1}^{t-1}(1 - \eta l_i^\tau)$$

2. Exponential weights (Hedge) Following a similar thought process as multiplicative weights:

$$x_i^t \propto \exp\left(-\eta \sum_{t=\tau}^{t-1} l_i^\tau\right)$$

We note that this approach is in fact equivalent to FTRL, in particular when finding the maximum entropy solution of:

$$x^t = \arg\min_{x \in \Delta(N)} \sum_{\tau=1}^{t-1} l_t(x) + R(x).$$

where $R(x)$ serves as an entropy regularization term:

$$R(x) = \frac{1}{\eta}\sum_{i=1}^{N} x_i \ln\left(\frac{1}{x_i}\right).$$

Observe $0 \leq R(x) \leq \frac{\ln(N)}{\eta}$.

5.2 Analysis of follow-the-regularized-leader (FTRL)

At its core, FTRL is implementing the idea of optimizing with stability.

We begin our analysis by first examining the *Be-the-Regularized-Leader* (BRL) approach. In this case, we assume that the initial round at $t = 0$ is dedicated to optimizing the regularization term:

$$l_0 = R(x).$$

By being the regularized leader, we "see" the loss l_t before choosing a new distribution x. In other words, we use x^{t+1} for round t. Consequently, by *using the next step to optimize the previous loss*, we perform better than any fixed choice of x. This leads to the inequality:

$$l_0(x^1) + \sum_{t=1}^{T} l_t(x^{t+1}) \leq l_0(x) + \sum_{t=1}^{T} l_t(x), \quad \forall x.$$

Referring to the definition of regret and using the previously derived inequality, we obtain:

$$\begin{aligned}\text{Regret} &= \sum_{t=1}^{T} l_t(x^t) - \sum_{t=1}^{T} l_t(x) \\ &\leq \sum_{t=1}^{T} \left(l_t(x^t) - l_t(x^{t+1})\right) + l_0(x) - l_0(x').\end{aligned}$$

The second term, $l_0(x) - l_0(x')$, is related to how well the sequence x^{t+1} performs compared to the fixed benchmark x. Using the assumption from BRL, we note that:

$$l_0(x) - l_0(x') = R(x) - R(x') \leq \frac{\ln(N)}{\eta}.$$

For now, we will ignore this term.

Taking a closer look at the first term, $\sum_{t=1}^{T}(l_t(x^t) - l_t(x^{t+1}))$, we see that this captures the stability of the updates in the algorithm. We note that if x^t and x^{t+1} are exactly the same, this difference is zero, and we have achieved a perfectly stable algorithm. If the update rule of the algorithm is not drastic, this difference remains small. Conversely, if the updates between adjacent rounds are large, this difference will also be large.

As a consequence, we can use this observation to understand why *Follow-the-Leader* (FTL) is unstable. In FTL, rapidly swapping between different experts results in a large difference between consecutive terms, leading to instability.

5.2.1 Bounding Stability

Consider a lemma that

$$\sum_{t=1}^{T} l_t(x^t) - l_t(x^{t+1}) \leq 2\eta T$$

which implies for values up to a factor of $e^\eta \approx (1 + 2\eta)$ that

$$x_i^t \approx x_i^{t+1}$$

So, regret for FTRL:

$$\text{Regret} \leq 2\eta T + \frac{\ln(N)}{\eta}$$

By optimizing for η we find that the Regret would be minimum when $\eta = \sqrt{\frac{\ln(N)}{2T}}$. This means that regret will grown proportional to

$$Regret \leq O\left(\sqrt{T\ln(N)}\right)$$

5.3 Online convex optimization

Let's assume for $t = 1, \dots, T$:

- Learner chooses $x^t \in \mathcal{X}$ and that this is convex.
- Adversary presents loss function $l_t \to l_t(x^t)$, where l_t is convex, differentiable, and Lipschitz continuous.

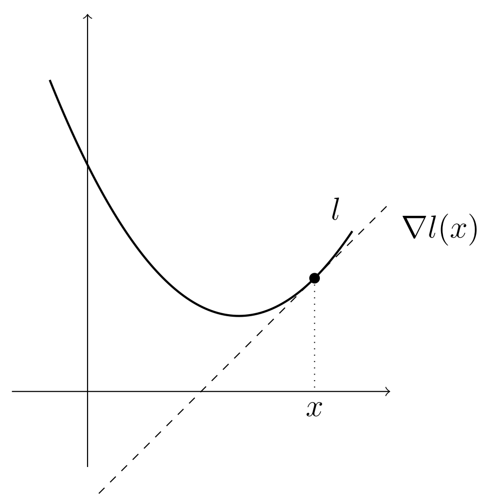

Figure 5.1: Assuming a l is a convex function, we can use gradient at point x

We can revise our FTRL algorithm by substituting the loss l_t with the gradient $\nabla l_t(x^t)$ at x_t.

$$x^t = \arg\min_{x \in \Delta(\mathcal{X})} \sum_{t=1}^{T-1} \langle \nabla l_t(x^t), x \rangle + R(x)$$

5.3.1 Special case: Online Gradient Descent

For a special case of defining $R(x)$ as the Euclidean Norm

$$R(x) = \frac{1}{2\eta}\|x\|_2^2 \quad , x \in \mathbb{R}^d$$

If we optimize w.r.t x

$$\begin{aligned} \sum_{\tau=1}^{t-1} \nabla l_t(x^t) + \frac{1}{\eta}x = 0 \\ x = -\eta \sum_{\tau=1}^{t-1} \nabla l_\tau(x^\tau) \end{aligned} \tag{5.1}$$

Which can be reformulated as gradient descent step

$$x^{t+1} = x^t - \eta \nabla l_t(x^t)$$