CS269: Machine Learning Theory
Lecture 11: Regularization in Online Learning

November 1, 2010

Lecturer: Jennifer Wortman Vaughan
Scribe: Jerry Weng, Li Yang Ku

Today we are going to continue on the topic of adversarial online learning. In particular, we will examine an alternative view of online learning based on regularization. We will show that by using regularization, algorithms become more stable, and will derive a regret bound for algorithms of this form.

1 Online Convex Optimization

Before jumping into regularization we first review the online convex optimization setting. The general regret bound that we prove will hold in this very general setting, though the particular example we consider later this lecture will be in the simple expert advice setting.

- For each round t
  - Algorithm chooses $w_t \in K$, where K is a convex set.
  - Adversary/Nature chooses the convex loss function l_t
  - Algorithm suffers loss $l_t(w_t)$ and observes the full function l_t
- Regret: $\sum_{t=1}^{T} l_t(w_t) - \min_{w \in K} \sum_{t=1}^{T} l_t(w)$

The function l_t chosen by the adversary each round can be an arbitrary convex loss function with bounded output. The results we prove will hold under the assumption that the losses are bounded in $[0, 1]$. We can think of regret as the accumulated loss by comparing our result with the best possible result. Since the adversary can simply set the loss function to be 1 everywhere on each round, it would be uninteresting to try to find an algorithm that minimizes absolute loss. Therefore, we try to find the algorithm that minimizes regret instead.

2 Follow the Leader

To show how regularization works, we first show an example where an algorithm that looks very reasonable fails in an adversarial setting. We will show that by adding a regularizer, we can achieve a much better result.

Consider the simple expert (linear loss) setting. Intuitively, it might seem to be a good idea to choose the distribution $\vec{w}$ with the best performance on previously observed data. This is equivalent to choosing the expert with the minimum empirical loss.

$$\begin{aligned}\vec{w}_t &= \arg\min_{\vec{w}\in\Delta_n} \sum_{s=1}^{t-1} \vec{l}_s \cdot \vec{w} \\ &= \arg\min_{\vec{w}\in\Delta_n} \vec{L}_{t-1} \cdot \vec{w}.\end{aligned}$$

However, an adversary knowing our algorithm can choose a bad sequence to fool our algorithm. Consider the following sequence where there are only two options for each step.

$$l_1 = (\frac{1}{2}, 0), l_2 = (0, 1), l_3 = (1, 0), l_4 = (0, 1), \cdots$$

Since our algorithm sets all the weight on the best expert in history, the weight sequence generated by our algorithm becomes:

$$w_1 = (\frac{1}{2}, \frac{1}{2}), w_2 = (0, 1), w_3 = (1, 0), w_4 = (0, 1), \cdots$$

We can easily see that our algorithm is tricked by the adversary. In fact, the best expert for this sequence is either one of the two experts, which each have $T/2$ loss. Thus, the regret for this algorithm would be:

$$Regret : T - \frac{T}{2} = \frac{T}{2}$$

which is linear in T.

3 Follow the Regularized Leader

3.1 Observation

In the previous section, the “Follow the Leader” algorithm is proved to be bad in an adversarial setting. In this section we will show a slightly modified algorithm, namely, “Follow the Regularized Leader.”

$$\vec{w}_t = \arg\min_{\vec{w}\in K} \left[\eta \sum_{s=1}^{t-1} l_s(\vec{w}) + R(\vec{w}) \right] \quad (1)$$

Here $R(\vec{w})$ is the regularizer, which is a convex function of $\vec{w}$, and $\eta > 0$ is a parameter. By adding this term we can get a more stable algorithm which doesn’t switch wildly. In the following sections, we will try to prove the regret bound for the Follow-the-Regularized-Leader algorithm.

3.2 Be-the-Regularized-Leader Lemma

Before stating the regret bound for Follow the Regularized Leader, we prove the following useful lemma. This lemma can be viewed as a regret bound for a hypothetical algorithm that chooses the point $\vec{w}$ minimizing $\eta \sum_{s=1}^{t} l_s(\vec{w}) + R(\vec{w})$ at each time t. Note that it is not actually possible to run such an algorithm since l_t is not known to the algorithm when this point is chosen. However, we will be able to use this bound to derive the regret bound for FTRL.

Lemma 1 (Be-the-Regularized-Leader Lemma)**.** *For any* $\vec{w} \in K, \eta > 0$,

$$\sum_{t=1}^{T} l_t(\vec{w}_{t+1}) - \sum_{t=1}^{T} l_t(\vec{w}) \leq \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1))$$

Proof: This prove is done by induction.

For $T = 0$, the LHS of the equation is 0. By definition:

$$\vec{w}_1 \quad = \arg\min_{\vec{w} \in K} [\eta \cdot 0 + R(\vec{w})]$$

Therefore, for any $\vec{w}$,

$$RHS = \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1)) \geq 0 = LHS$$

Therefore the lemma holds for $T = 0$.

Now suppose the equation holds for $T - 1$, then:

$$\forall \vec{w} \in K, \sum_{t=1}^{T-1} l_t(\vec{w}_{t+1}) - \sum_{t=1}^{T-1} l_t(\vec{w}) \leq \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1))$$

$$\forall \vec{w} \in K, \sum_{t=1}^{T-1} l_t(\vec{w}_{t+1}) + \frac{1}{\eta}R(\vec{w}_1) \leq \sum_{t=1}^{T-1} l_t(\vec{w}) + \frac{1}{\eta}R(\vec{w})$$

Since the inequality holds for any $\vec{w}$ we replace it with $\vec{w}_{T+1}$

$$\sum_{t=1}^{T-1} l_t(\vec{w}_{t+1}) + \frac{1}{\eta}R(\vec{w}_1) \leq \sum_{t=1}^{T-1} l_t(\vec{w}_{T+1}) + \frac{1}{\eta}(R(\vec{w}_{T+1}))$$

Then adding $l_T(\vec{w}_{T+1})$ to both sides we get:

$$\sum_{t=1}^{T} l_t(\vec{w}_{t+1}) + \tfrac{1}{\eta}R(\vec{w}_1) \quad \leq \sum_{t=1}^{T} l_t(\vec{w}_{T+1}) + \tfrac{1}{\eta}(R(\vec{w}_{T+1}))$$

By the definition, $\vec{w}_{T+1}$ is the $\vec{w} \in K$ that minimizers the RHS of the equation. Therefore, any other $\vec{w}$ can only increase the value of RHS. Therefore,

$$\begin{aligned} \sum_{t=1}^{T} l_t(\vec{w}_{t+1}) + \tfrac{1}{\eta}R(\vec{w}_1) \quad &\leq \sum_{t=1}^{T} l_t(\vec{w}_{T+1}) + \tfrac{1}{\eta}(R(\vec{w}_{T+1})) \\ &\leq \sum_{t=1}^{T} l_t(\vec{w}) + \tfrac{1}{\eta}(R(\vec{w})) \end{aligned}$$

for any $\vec{w}$. Hence the inequality also holds for T. Therefore by induction we proved Lemma 1.

□

3.3 Follow-the-Regularized-Leader regret bound.

Now by using the Be-the-Regularized-Leader Lemma we just proved, we can get an upperbound on the regret function.

From the Be-the-Regularized-Leader Lemma, adding $\sum_{t=1}^{T} l_t(\vec{w}_t)$ on both sides of the equality we can get:

$$\begin{aligned} \sum_{t=1}^{T} l_t(\vec{w}_t) - \sum_{t=1}^{T} l_t(\vec{w}) &\leq \sum_{t=1}^{T} l_t(\vec{w}_t) - \sum_{t=1}^{T} l_t(\vec{w}_{t+1}) + \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1)) \\ &= \sum_{t=1}^{T} (l_t(\vec{w}_t) - l_t(\vec{w}_{t+1})) + \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1)) \end{aligned} \tag{2}$$

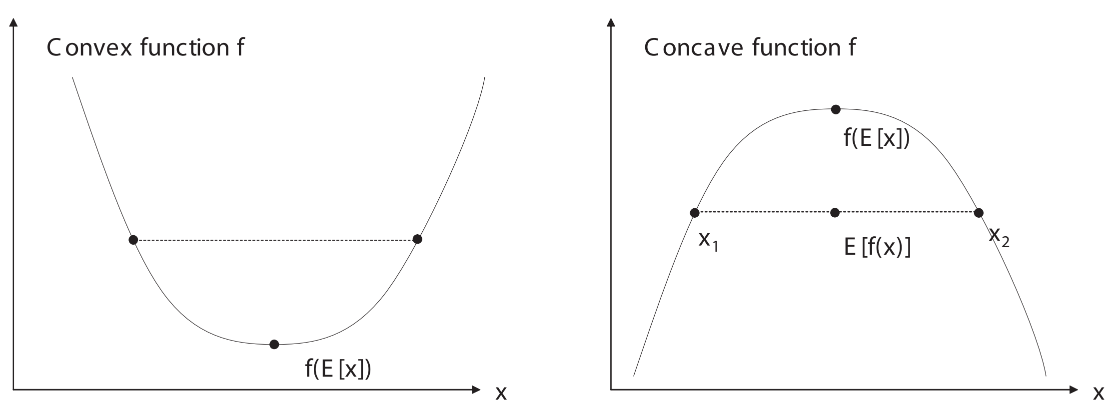
We can see that the left hand side of the inequality is the regret function, and the right hand side is its upperbound. Therefore, our goal would be to find the upperbound of $\sum_{t=1}^{T}(l_t(\vec{w}_t) - l_t(\vec{w}_{t+1}))$ and $R(\vec{w}) - R(\vec{w}_1)$.

3.4 Jensen's Inequality

In order to continue deriving the upperbound of our regret, we have to first introduce Jensen's Inequality. Jensen's Inequality is very helpful in machine learning.

Theorem 1. *(Jensen's Inequality) For any convex function f,* $f(E[x]) \leq E[f(x)]$*; for any concave function,* $E[f(x)] \leq f(E[x])$.

To remember which way the inequalities go, it is useful to keep in mind the following picture.



(a) convex graph (b) concave graph

3.5 Entropy

While the Follow the Regularized Leader bound holds for any convex loss function bounded in [0,1] for every round, we first turn our attention back to the simple expert (linear loss) setting again to see an example of how it can be used.

To find the upperbound of the regret function, we have to define our regularization term. The regularization term works by making the algorithm tend to choose a $\vec{w}$ that is more balanced and does not concentrate on a single expert. From the Follow the Leader algorithm, we can see that an algorithm that blindly chooses the best expert by past data could be easily fooled by an adversary; while, an algorithm that sets balanced weights to each expert would be less vulnerable.

Intuitively, using the negative entropy as the regularization term would be a quite adequate method for two reasons. First, the entropy gives a measure of how uniform the weights are, a higher entropy tend to make the weights less extreme. Second, the entropy has a clear bound which simplifies the process of finding the upperbound of the regret function.

First we introduce the entropy function:

$$H(\vec{w}) = \sum_{i=1}^{n} w_i \log \frac{1}{w_i} \qquad (\text{Define } 0 \cdot \log \frac{1}{0} = 0)$$

We can see that:

- When $\vec{w}$ is uniform $\rightarrow H(\vec{w}) = n \cdot \frac{1}{n} \log n = \log n$
- When $\vec{w}$ concentrates on one point $\rightarrow H(\vec{w}) = 1 \log 1 = 0$

Further more we want to show the fact that $\forall \vec{w} \in \Delta_n, 0 \leq H(\vec{w}) \leq \log n$

1. Since $0 \leq w_i \leq 1$, we can see that $w_i \log \frac{1}{w\,i} \geq 0$.
2. By using Jensen's inequality we mentioned in last section we can derive the following upper bound on entropy:

$$\begin{aligned} H(\vec{w}) = \sum_{i=1}^{n} w_i \log \frac{1}{w_i} &\leq \log \sum_{i=1}^{n} w_i \frac{1}{w_i} \\ &= \log \sum_{i=1}^{n} 1 = \log n \end{aligned}$$

Therefore we can see that $\forall \vec{w} \in \Delta_n, 0 \leq H(\vec{w}) \leq \log n$.

By using the negative entropy as our regularizer, we can derive an upperbound on the second term of equation (3).

$$\begin{aligned} &\Rightarrow R(\vec{w}) = -H(\vec{w}) = -\sum_{i=1}^{n} w_i \log \frac{1}{w\,i} \\ &\Rightarrow -\log n \leq R(\vec{w}) \leq 0 \\ &\Rightarrow \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1)) \leq \frac{1}{\eta} \log n \end{aligned} \tag{3}$$

3.6 Randomize Weighted Majority

Before deriving the upperbound of the first term in equation (2) to get a upperbound of our regret function, we will first try to prove that the Randomized Weighted Majority algorithm, which Jake talked about last week, turned out to be the same algorithm as Follow the Regularized Leader algorithm using a negative entropy regularizer. To prove this fact, we first find a lowerbound on $\eta \sum_{s=1}^{t-1} l_s(\vec{w}) + R(\vec{w})$ that holds for all $\vec{w} \in \Delta_n$, and then show that Randomized Weighted Majority chooses weights that minimize this expression.

$$\begin{aligned}
\eta \sum_{s=1}^{t-1} l_s(\vec{w}) + R(\vec{w}) &= \eta \sum_{s=1}^{t-1} \vec{l}_s \cdot \vec{w} - H(\vec{w}) && (4)\\
&= \eta \vec{L}_{t-1} \cdot \vec{w} - H(\vec{w})\\
&= \eta \sum_{i=1}^{n} L_{i,t-1} w_i - \sum_{i=1} w_i \cdot \log \frac{1}{w_i}\\
&= \sum_{i=1}^{n} w_i (\eta L_{i,t-1} - \log \frac{1}{w_i})\\
&= -\sum_{i=1}^{n} w_i \log(\frac{e^{-\eta L_{i,t-1}}}{w_i}) && (5)\\
&\geq -\log(\sum_{i=1}^{n} w_i \frac{e^{-\eta L_{i,t-1}}}{w_i})\\
&= -\log(\sum_{i=1}^{n} e^{-\eta L_{i,t-1}}) && (6)
\end{aligned}$$

Therefore $\forall \vec{w}, \eta \sum_{s=1}^{t-1} l_s(w) + R(w) \geq$ equation (6). Hence, if we can show that $\eta \sum_{s=1}^{t-1} l_s(w_t) + R(w_t)$ equals equation (6) for the weights $\vec{w}_t$ chosen by the Randomized Weighted Majority Algorithm, it must be the case that the Randomized Weighted Majority Algorithm chooses weights that minimize this expression.

Before continuing the proof, we first recall that the Randomized Weighted Majority Algorithm chooses a set of weights $\vec{w}_t$ for every round, where

$$w_{i,t} = \frac{e^{-\eta L_{1,t-1}}}{\sum_{j=1}^{n} e^{-\eta L_{j,t-1}}} \tag{7}$$

Now we will try to prove that the Randomized Weighted Majority algorithm is the algorithm that chooses $\vec{w}_t = \text{argmin}_{\vec{w} \in K} \left[\eta \sum_{s=1}^{t-1} l_s(\vec{w}) + R(\vec{w}) \right]$ for every round. By plugging equation (7) into equation (5) we get:

$$\eta \sum_{s=1}^{t-1} \vec{l}_s \cdot \vec{w}_t - H(\vec{w}_t) = -\sum_{i=1}^{n} w_{i,t} \log(\frac{e^{-\eta L_{i,t-1}}}{w_{i,t}}) \tag{8}$$

$$= -\sum_{i=1}^{n} w_{i,t} \log(\sum_{j=1}^{n} e^{-\eta L_{j,t-1}})$$

$$= -\log \sum_{i=1}^{n} e^{-\eta L_{i,t-1}} \tag{9}$$

Therefore, we can show that $\eta \sum_{s=1}^{t-1} l_s(w_t) + R(w_t)$ equals equation (6) for the weights w_t chosen by the Randomized Weighted Majority Algorithm. Hence, it must be the case that the Randomized Weighted Majority Algorithm chooses weights that minimize this expression, and we can conclude that the Randomized Weighted Majority algorithm is the Regularized Leader algorithm when a negative entropy regularizer is used.

3.7 Upperbound on regret function

Knowing the upperbound of the regularization term derived in section 3.5 and the closed form of $\vec{w}$ proved in section 3.6 we can therefore start deriving an upperbound on the regret function in equation (1). We will first introduce a Lemma, which we will prove in the next class.

Lemma 2. *For Randomized Weighted Majority, for all t,*

$$\vec{l_t} \cdot \vec{w}_t - \vec{l_t} \cdot \vec{w}_{t-1} \leq 2\eta$$

Assume Lemma 2 is true, we can then obtain the upperbound of the first term in equation (2). Together with the upperbound on the second term in equation (2) derived from equation (3), we can obtain an upperbound on the regret function:

$$\sum_{t=1}^{T} l_t(\vec{w}_t) - \sum_{t=1}^{T} l_t(\vec{w}) \leq \sum_{t=1}^{T} (l_t(\vec{w}_t) - l_t(\vec{w}_{t+1})) + \frac{1}{\eta}(R(\vec{w}) - R(\vec{w}_1))$$

$$\leq 2T\eta + \frac{1}{\eta}(\log n) \tag{10}$$

To obtain the optimal upperbound we choose the η that makes the two terms in equation (10) equal. Which is $\eta = \sqrt{\frac{\log n}{2T}}$. Plugging back η to equation (10) we get $Regret \leq 2\sqrt{2T \log n}$ which is proportion to $\sqrt{T}$ and consistent with the regret bound we proved for RWM in last week's classes.

In the next class, we will prove lemma 2.