Algorithms and Uncertainty, Winter 2017/18 Lecture 8 (4 pages)

No-Regret Learning: Experts

Instructor: Thomas Kesselheim

Today, we will come to a different online kind of online problem with different kinds of algorithms and performance measures.

As a motivating example, consider the following question of *binary classification.* You observe a sequence of samples from a data set and you have to classify them as "positive" or "negative". You make the choices one after the other, only after each choice you will get to know the true label for this sample. You can rely on a set of n classifiers that will each tell you their classification.

1 Majority Algorithm

Let us first consider the setting that one of the n classifiers is perfect and never makes any mistakes. The difficulty is: You do not know which it is.

A very simple and natural approach is the following *Majority* algorithm: Let S be the set of classifiers that have never been wrong so far. Follow the advice of the majority in S (with arbitrary tie breaking).

Observation 8.1. *If there is a perfect classifier, then the Majority algorithm makes at most* $\log_2 n$ *mistakes.*

Proof idea. Every time, the algorithm makes a mistake, at least $|S|/2$ of the classifiers in S are wrong. Therefore, in the following step, S will be at most half the size. As $1 \leq |S| \leq n$ in every step, the claim follows. □

2 Weighted Majority Algorithm

Let us now come to the general setting, in which each classifier makes a mistake every once in awhile. We would like to not make a lot more mistakes than the best among the n classifiers.

Of course, we cannot follow the above Majority rule because the set S will sooner or later be empty. Instead, we maintain for each classifier a weight w_i. Let $w_i(1) = 1$. If classifier i is correct in step t, then $w_i^{(t+1)} = w_i^{(t)}$, otherwise, if it is wrong, reduce it by setting $w_i^{(t+1)} = (1-\eta)w_i^{(t)}$, where $\eta \leq 1/2$ is a parameter of the algorithm. Our decision in step t is to follow the *weighted majority* of classifiers.

Theorem 8.2. *Weighted Majority makes at most* $(2+2\eta)\min_i m_i + 2\ln n/\eta$ *mistakes, where* m_i *is the number of mistakes that classifier* i *makes.*

Proof. Let $W^{(t)} = \sum_{i=1}^N w_i^{(t)}$ be the sum of weights in step t. Note that $W^{(t)}$ never increases as weights are only reduced. By the change of $W^{(t)}$, we can estimate how many mistakes Weighted Majority makes.

Consider a fixed step t, in which Weighted Majority makes a mistake. Let $U \subseteq [n]$ be the set of classifiers that are incorrect. Then, by definition $\sum_{i \in U} w_i^{(t)} \geq \sum_{i \notin U} w_i^{(t)}$, or equivalently $\sum_{i \in U} w_i^{(t)} \geq \frac{1}{2} \sum_i w_i^{(t)} = \frac{1}{2} W^{(t)}$.

For all $i \in U$, the algorithm updates the weight $w_i^{(t+1)} = (1-\eta)w_i^{(t)}$. For $i \notin U$, we have $w_i^{(t+1)} = w_i^{(t)}$. Therefore,

$$W^{(t+1)} = \sum_{i \in U} w_i^{(t+1)} + \sum_{i \notin U} w_i^{(t+1)} = \sum_{i \in U} (1-\eta) w_i^{(t)} + \sum_{i \notin U} w_i^{(t)} = W^{(t)} - \eta \sum_{i \in U} w_i^{(t)} \leq \left(1 - \frac{\eta}{2}\right) W^{(t)} \enspace .$$

Let M be the number of mistakes that the algorithm makes within the first T steps. By this observation, we have $W^{(T+1)} \leq \left(1 - \frac{\eta}{2}\right)^M W^{(1)} = \left(1 - \frac{\eta}{2}\right)^M n$.

Let m_i be the number of mistakes that classifier i makes within the first T steps. The algorithm is defined to set $w_i^{(T+1)} = (1-\eta)^{m_i} w_i^{(1)} = (1-\eta)^{m_i}$. Also $W^{(T+1)} \geq w_i(T+1)$.

Combining these two bounds, we get

$$(1-\eta)^{m_i} \leq W^{(T+1)} \leq \left(1 - \frac{\eta}{2}\right)^M n \enspace .$$

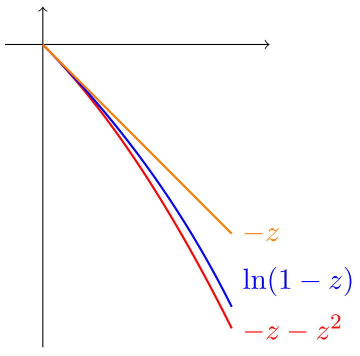
Let us take the logarithm on both sides

$$m_i \ln(1-\eta) \leq M \ln\left(1 - \frac{\eta}{2}\right) + \ln n \enspace .$$

In order to simplify, we will now use the following estimation

$$-z - z^2 \leq \ln(1-z) \leq -z \enspace , \tag{1}$$

which holds for every $z \in [0, \frac{1}{2}]$.



Therefore

$$m_i(-\eta - \eta^2) \leq M\left(-\frac{\eta}{2}\right) + \ln n \enspace ,$$

or equivalently

$$M \leq (2 + 2\eta) m_i + 2\frac{\ln n}{\eta} \enspace .$$

$\square$

3 Randomized Weighted Majority Algorithm

Now, we will see that we can actually do much better by using randomization. Not only the guarantee will be better but we also cover a more general setting. Instead of having binary classifiers, we now have arbitrary *experts*, which give us a piece of advice for every round. We choose one of these experts and follow her advice. In particular, the advice could simply be the positive or negative label. Afterwards, we get to know how good each of these experts performed in this round.

We will consider a sequence of cost vectors $(\ell_i^{(t)})_{i\in[n],t\in[T]}$, $\ell_i^{(t)} \in [0,1]$ for all i and t. In step t, we choose an expert I_t at random, then we get to know $\ell_1, \ldots, \ell_n$ and incur costs ℓ_{I_t}.

The idea of *Randomized Weighted Majority* is generally the same as for Weighted Majority. It maintains weights $w_i^{(t)}$, which are updated depending on the performance in the previous round. Instead of using a majority vote, we now interpret them a probability distributions.

Let $\eta \in (0, \frac{1}{2}]$; we will choose η later.

- Initially, set $w_i^{(1)} = 1$, for every $i \in [n]$.
- At every time t,
  - Let $W^{(t)} = \sum_{i=1}^N w_i^{(t)}$;
  - Choose strategy i with probability $p_i^{(t)} = w_i^{(t)}/W^{(t)}$;
  - Set $w_i^{(t+1)} = w_i^{(t)} \cdot (1-\eta)^{\ell_i^{(t)}}$.

Theorem 8.3 (Littlestone and Warmuth, 1994). *Randomized Weighted Majority, for any sequence of cost vectors from* $[0,1]$, *guarantees*

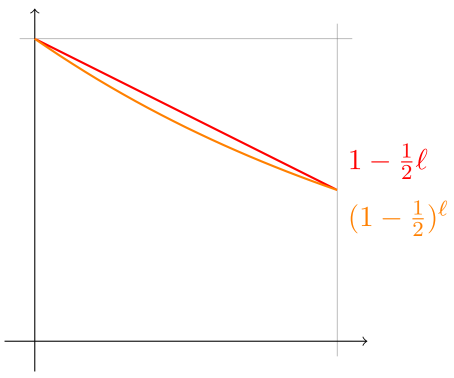
$$L_{RWM}^{(T)} \leq (1+\eta)L_i^{(T)} + \frac{\ln n}{\eta} ,$$

where $L_i^{(T)} = \sum_{t=1}^T \ell_i^{(t)}$ *is the sum of costs of expert* i *and* $L_{RWM}^{(T)} = \sum_{t=1}^T \sum_{i=1}^n p_i^{(t)} \ell_i^{(t)}$ *is the expected sum of costs of RWM.*

Proof. Let us analyze how the sum of weights $W^{(t)}$ decreases over time. It holds

$$W^{(t+1)} = \sum_{i=1}^N w_i^{(t+1)} = \sum_{i=1}^N w_i^{(t)} (1-\eta)^{\ell_i^{(t)}} .$$

Observe that $(1-\eta)^\ell = (1-\ell\eta)$, for both $\ell = 0$ and $\ell = 1$. Furthermore, $(1-\eta)^\ell$ is a convex function in ℓ. For $\ell \in [0,1]$ this implies $(1-\eta)^\ell \leq (1-\ell\eta)$.



This gives us

$$W^{(t+1)} \leq \sum_{i=1}^N w_i^{(t)}(1 - \ell_i^{(t)}\eta) = W^{(t)} - \eta \sum_{i=1}^N w_i^{(t)} \ell_i^{(t)} .$$

Let $\ell^{(t)}_{\text{RWM}}$ denote the expected cost of RWM in step t. It holds $\ell^{(t)}_{\text{RWM}} = \sum_{i=1}^{N} \ell^{(t)}_i w^{(t)}_i / W^{(t)}$. Substituting this into the bound for $W^{(t+1)}$ gives

$$W^{(t+1)} \leq W^{(t)} - \eta \ell^{(t)} W^{(t)} = W^{(t)}(1 - \eta \ell^{(t)}_{\text{RWM}}) \ .$$

As a consequence,

$$W^{(T+1)} \leq W^1 \prod_{t=1}^{T} (1 - \eta \ell^{(t)}_{\text{RWM}}) = N \prod_{t=1}^{T} (1 - \eta \ell^{(t)}_{\text{RWM}}) \ .$$

The sum of weights after step T can be upper bounded in terms of the expected costs of RWM. On the other hand, the sum of weights after step T can be lower bounded in terms of the costs of the best strategy as follows:

$$W^{(T+1)} \geq w_i^{(T+1)} = \left(w_i^1 \prod_{t=1}^{T} (1-\eta)^{\ell^{(t)}_i} \right) = \left((1-\eta)^{\sum_{t=1}^{T} \ell^{(t)}_i} \right) = (1-\eta)^{L^{(T)}_i} \ .$$

Combining the bounds and taking the logarithm on both sides gives us

$$L^{(T)}_i \ln(1-\eta) \leq (\ln n) + \sum_{t=1}^{T} \ln(1 - \eta \ell^{(t)}) \ .$$

Applying Equation (1), we get

$$\begin{aligned} L^{(T)}_i (-\eta - \eta^2) &\leq (\ln n) + \sum_{t=1}^{T} (-\eta \ell^{(t)}) \\ &= (\ln n) - \eta L^{(T)}_{\text{RWM}} \ . \end{aligned}$$

Finally, solving for $L^{(T)}_{\text{RWM}}$ gives

$$L^{(T)}_{\text{RWM}} \leq (1+\eta) L^{(T)}_i + \frac{\ln n}{\eta} \ . \quad \square$$

Note that setting $\eta = \sqrt{\frac{\ln n}{T}}$ yields

$$L^{(T)}_{\text{RWM}} \leq \min_i L^{(T)}_i + 2\sqrt{T \ln n} \ .$$

We call $R^{(t)} = L^{(T)}_{\text{RWM}} - L^{(T)}_i$ the (external) regret of the algorithm on the sequence. An algorithm that guarantees that for any sequence $R^{(t)} = o(T)$ is called a no-external-regret algorithm.

Corollary 8.4. *The multiplicative-weights algorithm with $\eta = \sqrt{\frac{\ln n}{T}}$ has external regret at most $2\sqrt{T \ln n} = o(T)$ and hence is a no-external-regret algorithm.*

4 Extension

The result so far assumed that $\ell^{(t)}_i \in [0,1]$ for all i and t. This result can be extended to allow $\ell^{(t)}_i \in [0,\rho]$ for all i and t: We divide all observed costs by ρ and feed them into the RWM. We then get

$$\frac{1}{\rho} L^{(T)}_{\text{RWM}} \leq \frac{1}{\rho} \min_i L^{(T)}_i + 2\sqrt{T \ln n} \ .$$

So, the new algorithm's regret is at most $2\rho\sqrt{T \ln n}$.