CS6781 - Theoretical Foundations of Machine Learning

Lecture 12: Introduction to Online Learning II

03/03/20

Lecturer: Nika Haghtalab *Readings: Blum and Mansour, Chapter 4*
Scribe: Seth Strimas-Mackey and Eric Landgrebe

1 Analysis of Randomized Weighted Majority (RWM)

Last time we introduced the Randomized Weight Majority (RMW) algorithm, which attempts to learn in an online setting against an adaptive adversary from a finite hypothesis class (equivalently set of experts) $\mathcal{H}$ of size n. Also recall that we denote the cost occured by expert i at time step t by $c^t(i) \in [0, 1]$.

Algorithm 1 Randomized Weighted Majority (ϵ)

$w_1^1, w_2^1, ..., w_n^1 = 1$
for $t = 1, 2, 3, ...$ **do**
  $W^t = \sum_{i=1}^{n} w_i^t$
  for $i \in [n]$, $p^t(i) = \frac{w_i^t}{W^t}$
  sample i^t from distribution given by $p^t = (p_1, \ldots, p_n)$
  for all $i : w_i^{t+1} \leftarrow w_i^t \cdot (1-\epsilon)^{c^t(i)}$
end for

Theorem 1.1. *For any $T > 0$ with an adaptive adversary, for the RMW algorithm:*

$$\mathbb{E}\left[\sum_{t=1}^{T} c^t(i^t)\right] \leq \frac{1}{1-\epsilon}\mathbb{E}\left[\min_{i\in[n]} \sum_{t=1}^{T} c^t(i)\right] + \frac{1}{\epsilon}\ln(N),$$

where both of the expectations are taken over the (potential) randomization of the adversary and the algorithm.

Proof. We will think of W^t as the "total credibility" we assigned to the various hypotheses in $\mathcal{H}$, and will begin by lower-bounding this quantity with respect to the optimal hypothesis in hind-sight.
Let $C^T(i) = \sum_{t=1}^{T} c^t(i)$.
Note that for all $i \in [n]$, $w_i^{T+1} = (1-\epsilon)^{c^1(i)} \cdot (1-\epsilon)^{c^2(i)} \cdots (1-\epsilon)^{c^T(i)} = (1-\epsilon)^{C^T(i)}$

Letting $i^* = \arg\min_{i\in[n]} c^T(i)$ we have:

$$W^{T+1} = \sum_{i=1}^{n} w_i^{T+1} \geq w_{i^*}^{T+1} = (1-\epsilon)^{C^T(i^*)}$$

So taking the natural log and expectation of both sides we get:

$$\mathbb{E}\left[\ln(W^{T+1})\right] \geq \mathbb{E}\left[C^T(i^*)\cdot\ln(1-\epsilon)\right] \tag{1}$$

Now we will bound W^{t+1} from above.

$$\begin{aligned}
W^{t+1} &= \sum_{i=1}^{n}(1-\epsilon)^{c^t(i)}w_i^t \\
&\leq \sum_{i=1}^{n}(1-c^t(i)\cdot\epsilon)w_i^t && \text{(using } (1-\epsilon)^c \leq 1-c\epsilon\text{)} \\
&= W^t - \epsilon\sum_{i=1}^{n} c^t(i)\cdot w_i^t \\
&= W^t\left(1-\epsilon\sum_{i=1}^{n} c^t(i)\cdot p^t(i)\right) && \left(\text{using } p^t(i) = \frac{w_i^t}{W^t}\right) \\
&= W^t\left(1-\epsilon\mathbb{E}_{i^t\sim p^t}\left[c^t(i^t)|\vec{w}^t\right]\right) && \text{(by definition } i^t\sim p^t\text{)} \\
&\leq W^t e^{-\epsilon\mathbb{E}\left[c^t(i^t)|\vec{w}^t\right]} && \text{(using } (1-x)\leq e^{-x}\text{)} \\
&\leq N\cdot e^{\left(-\sum_{\tau=1}^{t}\epsilon\mathbb{E}[c^\tau(i^\tau)|\vec{w}^\tau]\right)}. && \text{(unraveling over past steps)}
\end{aligned}$$

Now taking an expectation of W^{T+1} we get:

$$\mathbb{E}\left[W^{T+1}\right] \leq Ne^{-\sum_{t=1}^{T}\epsilon\mathbb{E}\left[c^t(i^t)\right]}.$$

Taking the natural log of both sides (using that $\ln$ is concave) by Jensen's inequality we get:

$$\mathbb{E}\left[\ln(W^{T+1})\right] \leq \ln\left(\mathbb{E}\left[W^{T+1}\right]\right) \leq \ln(N) - \epsilon\left(\sum_{t=1}^{T}\mathbb{E}\left[c^t(i^t)\right]\right) \tag{2}$$

Combining Equation 1 and Equation 2, we get

$$\mathbb{E}\left[C^T(i^*)\cdot\ln(1-\epsilon)\right] \leq \ln(N) - \epsilon\left(\sum_{t=1}^{T}\mathbb{E}\left[c^t(i^t)\right]\right),$$

and rearranging,

$$\sum_{t=1}^{T}\mathbb{E}\left[c^t(i^t)\right] \leq \frac{1}{\epsilon}\ln(N) - \mathbb{E}\left[C^T(i^*)\right]\frac{\ln(1-\epsilon)}{\epsilon}$$

$$= \frac{1}{\epsilon}\ln(N) + \mathbb{E}\left[C^T(i^*)\right]\frac{1}{\epsilon}\ln\left(\frac{1}{1-\epsilon}\right)$$
$$\leq \frac{1}{\epsilon}\ln(N) + \mathbb{E}\left[C^T(i^*)\right]\left(\frac{1}{1-\epsilon}\right).$$

Here the last step uses the fact that $\frac{1}{\epsilon}\ln\left(\frac{1}{1-\epsilon}\right) \leq \frac{1}{1-\epsilon}$, completing the proof. □

2 Defining Regret

For an algorithm $\mathcal{A}$ and a sequence $\{x_t\}_{t=1}^T$, we define the *regret of* $\mathcal{A}$ *on* $\{x_t\}_{t=1}^T$ as

$$\text{Regret}\left(\mathcal{A}, \{x_t\}_{t=1}^T\right) = \mathbb{E}\left[\sum_{t=1}^T c^t(i^t) - \min_{i\in[n]}\sum_{i=1}^T c^t(i)\right],$$

where $\{i^t\}_{t=1}^T$ is the sequence of choices that $\mathcal{A}$ makes given the input sequence $\{x_t\}_{t=1}^T$, and $c^t(i)$ is the cost of making choice i on round t. The first term in the expectation is the total cost incurred by $\mathcal{A}$, whereas the second term is the minimal total cost incurred by any expert. Thus the regret measures how well $\mathcal{A}$ does relative to the best expert in hindsight.

An algorithm $\mathcal{A}$ is said to have *regret bound* R if for any adaptive adversarial sequence $x_1, \ldots, x_T$,

$$\text{Regret}\left(\mathcal{A}, \{x_t\}_{t=1}^T\right) \leq R.$$

We can re-formulate Theorem 1.1 above as a regret bound as follows. Using that $(1-\epsilon)^{-1} \leq 1+2\epsilon$ whenever $\epsilon \in (0, 1/2)$ and rearrange terms, we find

$$\begin{aligned}\text{Regret}\left(\texttt{RWM}(\epsilon), \{x_t\}_{t=1}^T\right) &\leq 2\epsilon \cdot \mathbb{E}\left[\min_{i\in[n]}\sum_{i=1}^T c^t(i)\right] + \frac{1}{\epsilon}\ln(n) \\ &= 2\epsilon \cdot \mathbf{OPT} + \frac{1}{\epsilon}\ln(n), \end{aligned} \tag{3}$$

where we define $\mathbf{OPT}$ to be the expected cost of the best expert in hindsight,

$$\mathbf{OPT} = \mathbb{E}\left[\min_{i\in[n]}\sum_{i=1}^T c^t(i)\right].$$

Choosing

$$\epsilon^* = \min\left\{\frac{1}{2}, \sqrt{\frac{\ln(n)}{2\cdot\mathbf{OPT}}}\right\},$$

we find

$$\text{Regret}\left(\texttt{RWM}(\epsilon), \{x_t\}_{t=1}^T\right) \leq 2\sqrt{2\mathbf{OPT}\cdot\ln(n)} = O\left(\sqrt{\mathbf{OPT}\cdot\ln(n)}\right).$$

If **OPT** is not known, ϵ^* cannot be computed to run the $\texttt{RWM}(\epsilon^*)$ algorithm. However, if we know the number of steps T that the algorithm will be run for, we can use the fact that in the worst case, a cost of 1 is payed during each of the T rounds, and so $\textbf{OPT} \leq T$. Then, Equation 3 is further bounded by $(2\epsilon T + \ln(n)/\epsilon)$, so choosing

$$\epsilon_T = \min\left\{\frac{1}{2}, \sqrt{\frac{\ln(n)}{2T}}\right\},$$

we find

$$\text{Regret}\left(\texttt{RWM}(\epsilon), \{x_t\}_{t=1}^T\right) \in O\left(\sqrt{T \cdot \ln(n)}\right).$$

This assumes that T is known in advance. On the other hand, if the total runtime T is also not known, we can first guess a runtime small $\tilde{T}$, run the algorithm with for this number of steps with the associated step size. If the actual time exceed $\tilde{T}$, we double the estimate to $2\tilde{T}$ and restart the algorithm. We continue this process (doubling the runtime each time) until the total number of steps is at least T. Since the regret is $O\left(\sqrt{2^i \cdot \ln(n)}\right)$ each time we run the algorithm, and we run the algorithm $O\left(\log(T)\right)$ rounds in total (since we double the runtime each time), we find an overall regret bound of

$$\sum_{i=1}^{\log T} O\left(\sqrt{2^i \cdot \ln(n)}\right) = O\left(\sqrt{T \cdot \ln(n)}\right).$$

The *average* regret satisfies

$$\frac{\text{Regret}\left(\texttt{RWM}, \{x_t\}_{t=1}^T\right)}{T} \to 0 \qquad \text{as } T \to 0,$$

In general, we refer to any online algorithm with average regret that converges to zero as $T \to 0$ as a *no-regret algorithm*.

3 Online Learnability

We now move on to formally define learnability in the online setting.

Definition 3.1. Algorithm $\mathcal{A}$ learns class $\mathcal{H}$ in the online adversarial setting if there is a function $T_{\mathcal{H}} : (0,1) \to \mathbb{N}$ such that for any $\epsilon \in (0,1)$ and any $T \geq T_{\mathcal{H}}(\epsilon)$, the regret of algorithm $\mathcal{A}$ for any online sequence of length T satisfies

$$\frac{\text{Regret}\left(\mathcal{A}, \{x_t\}_{t=1}^T\right)}{T} \leq \epsilon.$$

In words, the regret is sublinear in T.

Using this definition, the regret bound for the RWM algorithm gives the following corollary for finite hypothesis classes.

Corollary 3.2. *If $\mathcal{H}$ is a finite hypothesis class, there exists an algorithm that learns it with*

$$T_{\mathcal{H}}(\epsilon) \in O\left(\frac{\ln(|\mathcal{H}|)}{\epsilon^2}\right).$$

What about for infinite hypothesis classes? In the PAC setting we know that if VCDim($\mathcal{H}$) $\leq d$, then $\mathcal{H}$ can be PAC-learned with $m_{\mathcal{H}}(\epsilon, \delta)$ samples, where

$$m_{\mathcal{H}}(\epsilon, \delta) = \Theta\left(\frac{1}{\epsilon^2}\left(d + \ln(1/\delta)\right)\right).$$

Can we also learn $\mathcal{H}$ with finite VC dimension in the online setting? It turns out the answer is no. To show this, let us consider $\mathcal{H}$ that is a class of threshold functions in 1-dimension:

$$\mathcal{H} := \{h_a : a \in [0, 1]\},$$

where $h_a(x) = \mathbf{1}(x \geq a)$. This class has VC dimension 1. We will first construct an adversarial sequence $(x_1, y_1), \ldots, (x_T, y_T)$ such that number of mistakes made by any algorithm is equal to T, but there exists $h^* \in \mathcal{H}$ that is consistent with the sequence.

1. Choose $x_1 = 1/2$ and let $\hat{y}_1$ be the guess of the algorithm for the label of x_1.
2. Choose y_1, the true label, to be the opposite of $\hat{y}_1$, i.e. $y_1 = 2\mathbf{1}(\hat{y}_1 = 0) - 1$. The algorithm thus makes a mistake on the first round.
3. For every subsequent round t, we set

$$x_t = x_{t-1} - \alpha_{t-1}\left(\frac{1}{2}\right)^t,$$

where $\alpha_t = -1$ if $y_t = 0$ and $\alpha_t = 1$ when $y_t = 1$. See the learner's prediction $\hat{y}_t$ and set $y_t = 2\mathbf{1}(\hat{y}_t = 0) - 1$ to be the opposite of the algorithm's guess, so the algorithm makes a mistake on it.

Note that there is always a halfpspace that is consistent with the set of generated samples. Therefore, in the mistake bound model any algorithm will have an ∞ mistake bound.

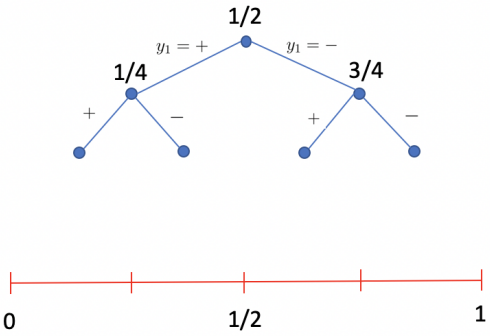

Figure 1: Illustration of sequence $(x_1, y_1), \ldots, (x_T, y_T)$.

For the online adversarial setting introduced above, note that (x_t, y_t) can depend on the history $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$, and the predictions $\hat{y}_1, \ldots, \hat{y}_{t-1}$, but cannot depend on the prediction $\hat{y}_t$ at time step t. To avoid this in the construction above, instead of choosing y_t to be the opposite label as $\hat{y}_t$, choose y_t uniformly at random from $\{0, 1\}$ and define x_t in the same way as before. Now at every time step any algorithm will make a mistake with probability $1/2$, but there still exists $h^* \in \mathcal{H}$ that is consistent with the sequence. Thus,

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}(\hat{y}_t \neq y_t) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{1}(h(x_t) \neq y_t)\right] = \frac{T}{2}.$$

This shows that $\mathcal{H}$ is not learnable in the online adversarial setting by any algorithm, even though it has VC dimension of 1. We thus see that online learnability is inherently different from offline learnability, and will require a notion other than VC dimension to be characterized.