COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Yanxi Chen

Lecture #15
April 1, 2019

1 Recap

We consider the problem of online learning, and more specifically, learning from experts. Denote by N the number of experts. The problem setting is as follows: for $t = 1, 2, \ldots, T$,

- Each expert predicts $\xi_i \in \{0, 1\}, i \in [N]$.
- The learner predicts $\hat{y} \in \{0, 1\}$.
- The learner observes true label $y \in \{0, 1\}$ (and makes a mistake if $y \neq \hat{y}$).

Note that in online learning, training and testing are mixed together. Also, compared to PAC learning we have studied before, fewer statistical assumptions are made in online learning, and we are considering an adversarial (or worst-case) setting.

In the previous lecture, we assumed that there exists (at least) one "perfect" expert that never makes mistakes. We learned about the halving algorithm: at each step, $\hat{y}$ is the (unweighted) majority vote of all surviving experts, and after y is revealed, we eliminate all experts that made a mistake ($\xi_i \neq y$). By the nature of the halving algorithm, when the learner makes one mistake, the number of surviving experts is reduced at least by half, which implies that the total number of mistakes m made by the learner satisfies $m \leq \lg N$ (where, as usual, lg denotes base-2 logarithm).

2 Online Learning vs. PAC Learning

We can consider a similar problem that can be viewed as an online analog of PAC learning. Denote by $\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$ the (finite) hypothesis class, and $c \in \mathcal{H}$ the target hypothesis. Then for each round,

- A data point x is observed.
- The learner predicts $\hat{y} \in \{0, 1\}$.
- The true label $y = c(x)$ is observed.

Here, the adversary chooses target c (before learning starts), as well as the data points x. We can treat this problem as a special case of learning with expert advice. In particular, we can view each hypothesis $h_i, i \in [N]$ as an expert, whose prediction on each round is set to the label given by the corresponding hypothesis so that $\xi_i = h_i(x)$. Then the number of experts is $N = |\mathcal{H}|$; also, $c \in \mathcal{H}$ implies that there exists a perfect expert. So, using the halving algorithm, we get an upper bound on the number of mistakes of

$$\lg N = \lg |\mathcal{H}|.$$

The right-hand side, $\lg |\mathcal{H}|$, can be seen as a natural complexity measure of the hypothesis class $\mathcal{H}$, which is similar to what we have seen at the beginning of our study of PAC learning.

It turns out that we can lower bound the number of mistakes by the VC-dimension of $\mathcal{H}$. Define

$$M_A(\mathcal{H}) = \max\ (\#\text{ mistakes of } A),$$

where A is a deterministic algorithm, and the maximum is taken over the adversary and measures the number of mistakes that A will make for the problem above for hypothesis space $\mathcal{H}$. Also define

$$\text{opt}(\mathcal{H}) = \min_A M_A(\mathcal{H})$$

to be the best-possible mistake bound for any deterministic algorithm A on hypothesis space $\mathcal{H}$.

Theorem 1.

$$\text{VCdim}(\mathcal{H}) \leq \text{opt}(\mathcal{H}) \leq M_{\text{halving}}(\mathcal{H}) \leq \lg |\,\mathcal{H}\,|.$$

Proof. The upper bound achieved by using the halving algorithm as A has been proved before, so it remains to prove the lower bound. We fix the deterministic algorithm A. Also denote $d = \text{VCdim}(\mathcal{H})$; this implies that there exist d points, $S = \{x_1, \ldots, x_d\}$, which are shattered by $\mathcal{H}$, i.e. any possible labeling on S can be realized by $\mathcal{H}$. Intuitively, this suggests that seeing labels for part of these d points actually reveals no information about labels of the remaining points. To make this precise, let us restate the problem of online learning from the perspective of the adversary. First, the adversary chooses $c \in \mathcal{H}$ (before learning happens), and then for $t = 1, \ldots, d$,

- The adversary presents $x_t \in S$ to the learner, i.e. the algorithm A.
- A makes prediction $\hat{y}_t$.
- The adversary reveals $y_t = c(x_t) \neq \hat{y}_t$, and A makes a mistake.

The reason why $y_t = c(x_t) \neq \hat{y}_t, \forall t \in [d]$ is possible is because, in our setting, the algorithm A is *deterministic* and known to the adversary; therefore, before learning happens, the adversary can "simulate" A and know how A will label data points in S, and then pick $c \in \mathcal{H}$ accordingly, so that $c(x_t) \neq \hat{y}_t, \forall t \in [d]$. In this sense, the adversary is definitely not "cheating". (In comparison, in the PAC learning model we have studied before, the adversary can only choose the distribution D, and then data is randomly generated according to D.) By the process above, we have $M_A(\mathcal{H}) \geq d$, and since this is true for any deterministic A, we have $\text{opt}(\mathcal{H}) \geq d = \text{VCdim}(\mathcal{H})$. □

3 Weighted Majority Algorithm (WMA)

In the previous sections, we assumed that there exists at least one perfect expert, which can be unrealistic in the real world. From now on, we drop such an assumption. Note that the existence of a perfect expert guarantees that there will be at least one expert surviving all rounds of the halving algorithm, while without a perfect expert, it is most likely that the halving algorithm will end up with no surviving expert. A natural solution to solve this problem is that, in each round, we take a *weighted* majority vote and update weights for all experts (more precisely, decrease the weight of an expert if it makes a mistake in this round), instead of taking an unweighted majority and totally eliminating experts who make a mistake. This leads to the **Weighted Majority Algorithm (WMA)**, as stated in Algorithm 1.

Algorithm 1: Weighted Majority Algorithm (WMA)

Input : Parameter $\beta \in [0, 1)$

Initialize: weight of expert i, $w_i = 1, i \in [N]$;

for $t = 1, \ldots, T$ **do**

  $q_1 = \sum_{i:\xi_i=1} w_i, q_0 = \sum_{i:\xi_i=0} w_i$, where ξ_i is the prediction of expert $i \in [N]$;

  Predict $\hat{y} = 1$ if $q_1 > q_0$ and $\hat{y} = 0$ otherwise ;

  Observe y, and for $i \in [N]$, if $\xi_i \neq y$, then update $w_i \leftarrow w_i \ \beta$.

end

In the setting where there exists no perfect expert, perhaps the best we can hope for is that the number of mistakes is not larger than that of the best expert by too much. The theorem below provides such an upper bound.

Theorem 2.

$$\#\text{ mistakes of WMA} \leq a_\beta \ (\#\text{ mistakes of best expert}) + c_\beta \lg N, \tag{1}$$

where

$$a_\beta = \frac{\lg \frac{1}{\beta}}{\lg \frac{2}{1+\beta}}, \quad c_\beta = \frac{1}{\lg \frac{2}{1+\beta}}.$$

Remark 1. *One may notice that there is a trade-off between a_β and c_β. A list of their values with some particular choices of β is listed below.*

β	a_β	c_β
$1/2$	≈ 2.4	≈ 2.4
$\to 0$	∞	1
$\to 1$	2	∞

Proof. Define the total weight

$$W = \sum_{i=1}^{N} w_i.$$

Note that W and the other variables we are using should be indexed by round t, but we omit subscripts for notational simplicity; also note that $W = q_1 + q_0$. Initially $W = N$. On round t, suppose that the true label is $y = 0$. Then

$$\begin{aligned} W^{\text{new}} = \sum_i w_i^{\text{new}} &= \sum_{i:\xi_i=1} w_i \ \beta + \sum_{i:\xi_i=0} w_i \\ &= q_1 \ \beta + q_0 = q_1 \ \beta + W - q_1 = W - (1-\beta) \ q_1. \end{aligned}$$

Therefore, if WMA makes a mistake at round t ($\hat{y} = 1$), then

$$q_1 \geq q_0 \Rightarrow q_1 \geq W/2 \Rightarrow W^{\text{new}} \leq \frac{1+\beta}{2} \ W.$$

The analysis for the case when true label $y = 1$ is the same. After WMA makes m mistakes,

$$W \leq N \ \left(\frac{1+\beta}{2}\right)^m.$$

Algorithm 2: Randomized Weighted Majority Algorithm (RWMA)

Input : Parameter $\beta \in [0, 1)$

Initialize: weight of expert i, $w_i = 1, i \in [N]$;

for $t = 1, \ldots, T$ **do**

  Predict $\hat{y} = \xi_i$ with probability w_i/W, where $W = \sum_{i=1}^{N} w_i$;

  Observe y, and for $i \in [N]$, if $\xi_i \neq y$, then update $w_i \leftarrow w_i \ \beta$.

end

On the other hand, for each expert $i \in [N]$,

$$w_i = \beta^{L_i}, \quad \text{where } L_i = \#\text{ mistakes of expert } i.$$

Since $w_i = \beta^{L_i} \leq W \leq N \ (\frac{1+\beta}{2})^m$, we have

$$m \leq \frac{L_i \ \lg \frac{1}{\beta} + \lg N}{\lg \frac{2}{1+\beta}}.$$

Since this is true for any $i \in [N]$, by taking the minimum over i, we complete the proof. □

Another way to understand this theorem is to divide both sides of (1) by T (the total number of rounds), which gives

$$\frac{\#\text{ mistakes of WMA}}{T} \leq a_\beta \ \frac{\#\text{ mistakes of best expert}}{T} + c_\beta \frac{\lg N}{T}. \tag{2}$$

Note that $\frac{\lg N}{T} \to 0$ as $T \to \infty$; then the left-hand side is the "rate of mistake" for WMA, while the right-hand side is a_β times the "rate of mistake" of the best expert. Unfortunately, for any possible choice of β, the coefficient $a_\beta \geq 2$, which is not ideal since we hope for a coefficient 1 so that the learner will not be doing much worse than the best expert. It turns out that it is impossible to achieve a coefficient 1 with a deterministic algorithm, which, roughly speaking, is due to the limit of a deterministic algorithm in an adversarial setting. This motivates us to design a randomized algorithm, which is the topic of the next section.

4 Randomized Weighted Majority Algorithm (RWMA)

The **Randomized Weighted Majority Algorithm (RWMA)** is stated in Algorithm 2. The only difference between RWMA and WMA is that, in RWMA, we pick expert i with probability proportional to w_i, and follow the prediction of the chosen expert.

Since RWMA is randomized, the number of mistakes is also random. The following theorem provides an upper bound of the expected number of mistakes by RWMA.

Theorem 3.

$$\mathbb{E}[\#\text{ mistakes of RWMA}] \leq a_\beta \ (\#\text{ mistakes of best expert}) + c_\beta \ln N, \tag{3}$$

where

$$a_\beta = \frac{\ln \frac{1}{\beta}}{1-\beta}, \quad c_\beta = \frac{1}{1-\beta}.$$

Remark 2. *Notice that in* (3)*, the expectation is only over the randomness of the algorithm; everything else is still adversarial.*

Proof. Initially $W = N$. On round t, define

$$\ell = \Pr(\hat{y} \neq y) = \frac{\sum_{i:\xi_i \neq y} w_i}{W}.$$

(Again, we omit the index t.) Then

$$\begin{aligned} W^{\text{new}} &= \sum_{i:\xi_i \neq y} w_i^{\text{new}} + \sum_{i:\xi_i = y} w_i^{\text{new}} = \sum_{i:\xi_i \neq y} w_i \ \beta + \sum_{i:\xi_i = y} w_i \\ &= W \ \ell \ \beta + W - W\ell = W \ (1 - \ell(1-\beta)). \end{aligned}$$

Now we bring back the index t, and

$$W_{\text{final}} = N \ \prod_{t=1}^{T}(1 - \ell_t(1-\beta)) \leq N \ \prod_{t=1}^{T} e^{-\ell_t(1-\beta)} = N \ e^{-(1-\beta)\sum_{t=1}^{T}\ell_t}.$$

The analysis for a single expert is the same as in the theorem of WMA, and we have $W_{\text{final}} \geq w_i = \beta^{L_i}$, where L_i is the number of mistakes made by expert i; this is true for all $i \in [N]$. Therefore $\beta^{L_i} \leq W_{\text{final}} \leq N \ e^{-(1-\beta)\sum_{t=1}^{T}\ell_t}$, which gives

$$\sum_{t=1}^{T} \ell_t \leq \frac{(\ln \frac{1}{\beta}) \ \min_i L_i + \ln N}{1-\beta}.$$

Notice that the left-hand side is actually

$$\sum_{t=1}^{T} \ell_t = \sum_t \Pr(\hat{y}_t \neq y_t) = \sum_t \mathbb{E}[\mathbb{1}(\hat{y}_t \neq y_t)] = \mathbb{E}[\#\text{ mistakes of RWMA}],$$

which completes our proof. □

Given (3), one is tempted to pick $\beta \to 1$ so that $a_\beta \to 1$, which is our motivation to use a randomized algorithm, as stated at the end of the previous section. However, this will give $c_\beta \to \infty$; therefore, the choice of β is not so trivial. For example, if we know a priori that $\min_i L_i \leq K$ for some K (e.g. a trivial bound is $K = T$), then we can pick $\beta = 1/(1 + \sqrt{2(\ln N)/K})$, and after some algebra, this gives

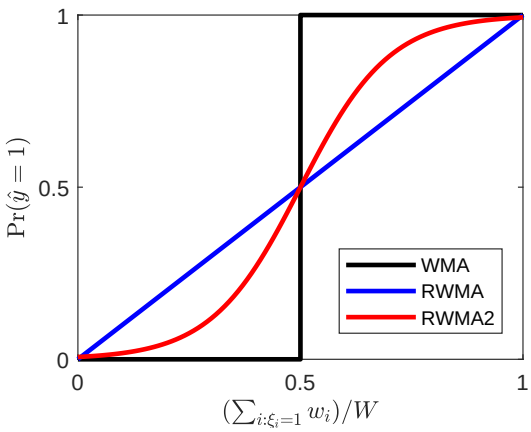
$$\#\text{ mistakes of RWMA} \leq \min_i L_i + \sqrt{2K \ln N} + \ln N,$$

or equivalently,

$$\frac{\#\text{ mistakes of RWMA}}{T} \leq \frac{\min_i L_i}{T} + \frac{\sqrt{2K \ln N}}{T} + \frac{\ln N}{T},$$

where the last two terms go to zero as $T \to \infty$, and RWMA achieves the same rate of mistake as the best expert.

As a final remark, let us view WMA and RWMA under a unified framework. The difference between WMA and RWMA is, in fact, their probabilities of choosing $\hat{y} = 1$ in terms of $Z := (\sum_{i:\xi_i=1} w_i)/W \in [0,1]$. For WMA, the probability is 0/1, using $Z = 0.5$ as the threshold; for RWMA, the probability is linear in Z, as shown in the figure below. This suggests that, by choosing a different function g for $\Pr(\hat{y} = 1|Z) = g(Z)$ (red line), we actually obtain a new randomized algorithm (which we call "RWMA2" in the figure below). For example, with a proper design of g, we can achieve $\#$ mistakes $\leq \min_i L_i + \sqrt{K \ln N} + (\lg N)/2$; this will be discussed in more detail in the next lecture.


1
0.5
0
0
0.5
1
$\Pr(\hat{y}=1)$
$(\sum_{i:\xi_i=1} w_i)/W$
WMA
RWMA
RWMA2