Lecture 13: Online LP & WMA I

17 February 2020

Lecturer: Dr. Kanat Tangwongsan   *Scribe: Kriangsak T. & Apivich H.*

1   Experts' Predictions Problem

Let's say we want to predict whether or not an event A will happen. In doing so, we will go about asking N number of experts to make T rounds of predictions. As an algorithm, in step t (where $t = 1, 2, \ldots, T$),

1. Each expert $i \in [N]$ advises: YES/NO

2. Aggregator predicts the outcome: YES/NO

3. An adversary (think of it as God who knows all an aggregator would predict) will determine for an whether an event will occur: YES/NO

4. Observe the outcome

In any predictions made, there can be the case when mistakes are made by those N experts. A mistake is when the predicted outcome we made does not match the actual outcome. Our goal is to minimise the number of mistakes made in one prediction.

Theorem 1.1. *If there is a perfect expert, then, there is an aggregator that makes at most $\log_2 N$ mistakes*

Proof. Think of this as how many mistakes have to made until the perfect expert is found. If a mistake has been made, at least $N/2$ experts were wrong. Therefore, there will be at most $\log_2 N$ mistakes made until the perfect expert is found. □

Theorem 1.2. *If the best expert makes m mistakes, then the aggregator makes $O(m(\log N) + \log N)$ mistakes.*

Proof. Note the following.

- Every run, imperfect experts make $\leq \log_2 N + 1$ mistakes and the perfect expert makes ≥ 1 mistake(s)

- There can be at most m runs: in the $m + 1$ run, the best expert makes no more mistake.

- By Theorem 1.1 all imperfect experts make $\leq \log_2 N$

Therefore, the total number of mistakes is $O(m(\log_2 N + 1) \log_2 N)$

$\square$

Remark: notice that the above bound is multiplicative of m, we will try to do better in trying to reduce it into an additive bound.

2 Weighted Majority Algorithm (WMA)

To continue with the expert example, we will try to come up with a few algorithms that will improve upon the bound from Theorem 1.2. What we will introduce is the idea of the Weighted Majority Algorithm (or WMA).

2.1 Initial Method

The idea of WMA is to trust the correct experts more than the incorrect ones. We will do this by weighing the answers from the more accurate experts more heavily, and weigh the less accurate experts less. We will do so by assigning weights to the different experts and reducing them when they get it wrong.

Initially, we define the weight for all the experts (in the 0th round) to be $w_i^{(0)} = 1$. For the next rounds, we define

$$w_i^{(t+1)} = \begin{cases} w_i^{(t)} & \text{if } i\text{th expert was correct,} \\ \frac{1}{2} w_i^{(t)} & \text{if } i\text{th expert was wrong.} \end{cases}$$

The prediction will be made and the weights adjusted. The algorithm is as noted below.

Algorithm 1 Weighted Majority Algorithm

1: **function** WMA
2:     Initialise $w_i^{(0)} = 1$ for all i in $1, 2, \ldots, N$
3:     **for** t **in** $1, 2, \ldots, T$ **do**
4:         Gather predictions (YES or NO) from each experts
5:         **if** $\sum\limits_{i \text{ say YES}} w_i^{(t)} > \sum\limits_{i \text{ say NO}} w_i^{(t)}$ **then**
6:             Predict YES and compare to the adversary
7:         **else**
8:             Predict NO and compare to the adversary
9:         Update each weights by $w_i^{(t+1)} \leftarrow \begin{cases} w_i^{(t)} & \text{if } i\text{th expert was correct in round } t, \\ (1 - \epsilon)w_i^{(t)} & \text{otherwise} \end{cases}$

We will now try to find how many times the algorithm will make the wrong decision. Let's define the potential function

$$\Phi^{(t)} = \sum_i^T w_i^{(t)}.$$

Notice that $\Phi^{(0)} = n$ and $\Phi^{(t+1)} \leq \Phi^{(t)}$.

Let's say a mistake was made at step t. Then,

$$
\begin{aligned}
\Phi^{(t+1)} &= \sum_{i=1}^T w_i^{(t+1)} \\
&= \sum_{i \text{ is right}} w_i^{(t+1)} + \sum_{i \text{ is wrong}} w_i^{(t+1)} \\
&= \sum_{i \text{ is right}} w_i^{(t)} + \frac{1}{2} \sum_{i \text{ is wrong}} w_i^{(t)} \\
&= \sum_{i \text{ is right}} w_i^{(t+1)} + (1 - \frac{1}{2}) \sum_{i \text{ is wrong}} w_i^{(t)} \qquad (*) \\
&= \Phi^{(t)} - \frac{1}{2} \sum_{i \text{ is wrong}} w_i^{(t)} \\
&\leq (1 - \frac{1}{4})\Phi^{(t)} \\
&= \frac{3}{4}\Phi^{(t)}
\end{aligned}
$$

From $(*)$, let's zoom in into what happens at this step. We can express the potential function as

$$\Phi^{(t)} = \sum_i^T w_i^{(t)}$$

$$= \sum_{\substack{i \text{ is right}}}^T w_i^{(t)} + \sum_{\substack{i \text{ is wrong}}}^T w_i^{(t)}$$

If we expand $(*)$, we see that

$$(*) = \underbrace{\sum_{\substack{i \text{ is right}}}^T w_i^{(t+1)} + \sum_{\substack{i \text{ is wrong}}}^T w_i^{(t)}}_{\Phi^{(t)}} - \frac{1}{2} \sum_{\substack{i \text{ is wrong}}}^T w_i^{(t)}$$

$$= \Phi^{(t)} - \frac{1}{2} \underbrace{\sum_{\substack{i \text{ is wrong}}}^T w_i^{(t)}}_{\leq \frac{1}{2}\Phi^{(t)}} \geq \frac{1}{2}\Phi^{(t)}$$

Now, enough with unattractive summations, we will analyse what would happen to the total number of mistakes (denoted by M) the aggregator makes if (perfect) expert i makes m_i mistakes, then

$$\left(\frac{1}{2}\right)^m = w_i^{(T)} \leq \Phi^{(T+1)} \leq \left(\frac{3}{4}\right)^M \Phi^{(0)}$$

which after taking log on both sides,

$$-m_i \leq M \log_2 \frac{3}{4} + \log_2 N$$

$$m_i \geq M \log_2 \frac{4}{3} - \log_2 N$$

$$M \leq \frac{1}{\log_2 \frac{4}{3}}(m_i + \log_2 N)$$

$$M \leq 2.41(m_i + \log_2 N)$$

We can see that the number of mistakes the algorithm will make is bounded by $2.41(m_i + \log_2 N)$.

2.2 Generalised WMA

We can make the WMA we have discussed more generalised. The way we do so is rather than halving the weight of the expert when they get it wrong, we instead reduce it by some other factor.

More specifically, we instead update the weights in Line 9 from above according to

$$w_i^{(t+1)} = \begin{cases} w_i^{(t)} & \text{if } i\text{th expert was correct in round } t, \\ (1-\epsilon)w_i^{(t)} & \text{otherwise} \end{cases}$$

where we define $0 < \epsilon < 1$. Similar to above, we can analyse how many incorrect guesses will be made by this method. As of above, we can define the potential to be $\Phi^{(t)} = \sum_i w_i^{(t)}$. By a similar analysis as above we can show that if the prediction is wrong in round t then $\Phi^{(t+1)} \le \left(1 - \frac{\epsilon}{2}\right)\Phi^{(t)}$.

This means that if we have an expert i who makes m_i mistakes, then after T turns,

$$w_i^{(T)} = (1-\epsilon)^{m_i}$$
$$\le \Phi^{(T+1)} \le \Phi^{(0)}\left(1 - \frac{\epsilon}{2}\right)^M = N\left(1 - \frac{\epsilon}{2}\right)^M$$

since $\Phi^{(t)}$ would have shrunken M times. This means that

$$(1-\epsilon)^{m_i} \le N\left(1 - \frac{\epsilon}{2}\right)^M$$
$$(1-\epsilon)^{m_i} \le Ne^{-\frac{M\epsilon}{2}} \qquad\qquad \text{since } 1 - x \le e^{-x}$$
$$m_i \ln(1-\epsilon) \le -\frac{M\epsilon}{2} + \ln N \qquad\qquad \text{by taking natural log on both side}$$
$$M \le \frac{2[\ln N - m_i \ln(1-\epsilon)]}{\epsilon}$$
$$M \le \frac{2}{\epsilon}[\ln N + m_i(\epsilon + \epsilon^2)] \qquad \text{since } \ln\left(\frac{1}{1-x}\right) \le x + \frac{x^2}{2} \le x + x^2$$
$$M \le 2m_i(1+\epsilon) + \frac{2\ln N}{\epsilon}$$

which means that the number of mistakes that we will make with our predictions is bounded by $2m_i(1+\epsilon) + O\left(\frac{\ln N}{\epsilon}\right)$. Note that if m_i^* is the number of mistakes made by the best expert, then M is bounded by $2m_i^*(1+\epsilon) + O\left(\frac{\ln N}{\epsilon}\right)$.

While this algorithm is good, there is a factor of 2 in the bound which cannot be gotten rid of. The reason for this is in the fact that our algorithm is deterministic. Imagine if we have two experts, one who always answers true to every case, and the other who answers false to every case. Then, if we make M mistakes, then one of the experts will be incorrect at most $M/2$ times. We can see that there is a factor of 2 between the number of times we get it wrong, and that the expert gets it wrong.

3   Randomised WMA

We can improve upon WMA from above by adding an element of randomess into it. The new algorithm will be as follows.

Algorithm 2 Randomised WMA

1: **function** RANDOMISEDWMA
2:      Initialise $w_i^{(0)} = 1$ for all i in $1, 2, \ldots, N$
3:   **for** t **in** $1, 2, \ldots, T$ **do**
4:          Gather predictions (YES or NO) from each experts
5:          $p \leftarrow \begin{cases} \text{NO} & \text{with probability } \dfrac{\sum\limits_{i \text{ say NO}} w_i^{(t)}}{\Phi^{(t)}} \\[2em] \text{YES} & \text{with probability } \dfrac{\sum\limits_{i \text{ say YES}} w_i^{(t)}}{\Phi^{(t)}} \end{cases}$
6:          Predict p and compare to the adversary
7:          Update each weights by $w_i^{(t+1)} \leftarrow \begin{cases} w_i^{(t)} & \text{if } i\text{th expert was correct in round } t, \\ (1-\epsilon)w_i^{(t)} & \text{otherwise} \end{cases}$

We can see that the algorithm above is exactly the same as in Algorithm 1, except for the fact that the prediction part (Lines 5-8) is replaced with a random decision process similar to a biased coin. This algorithm, however, is expected to result in lower number of incorrect predictions.

Theorem 3.1. *Let M be the number of wrong guesses made by* RANDOMISEDWMA. *Then,* $\mathbb{E}[M] = m_i(1 + \epsilon) + O\left(\frac{\ln N}{\epsilon}\right)$.

Proof. First, we define

$$F^{(t)} = \frac{\sum\limits_{i \text{ is wrong}} w_i^{(t)}}{\Phi^{(t)}},$$

which simply represents the total weight of the experts who got their answer wrong. This means that the probability that RANDOMISEDWMA gets the answer wrong in round t is $F^{(t)}$. By linearity of expectations, this means that $\mathbb{E}[M] = \sum_{t=1}^{T} F^{(t)}$.
We will now show a useful result. We can see that

$$\Phi^{(t+1)} = \underbrace{\sum_{i \text{ is right}} w_i^{(t)}}_{\text{if expert right, weight stays the same}} + \underbrace{(1-\epsilon) \sum_{i \text{ is wrong}} w_i^{(t)}}_{\text{if expert wrong, multiply factor}}$$

$$= \Phi^{(t)}[(1 - F^{(t)}) + (1-\epsilon)F^{(t)}]$$

$$= \Phi^{(t)}(1 - \epsilon F^{(t)}).$$

We can therefore see that if expert i makes m_i mistakes, then

$$w_i^{(T)} = (1 - \epsilon)^{m_i}, \text{ and}$$

$$w_i^{(T)} \leq \Phi^{(T)}$$

$$= \left(1 - \epsilon F^{(T-1)}\right)\Phi^{(T-1)}$$

$$= \left(1 - \epsilon F^{(T-1)}\right)\left(1 - \epsilon F^{(T-2)}\right)\Phi^{(T-2)}$$

$$\vdots$$

$$\leq \left(\prod_{t=1}^{T}(1 - \epsilon F^{(t)})\right)\Phi^{(0)}$$

$$= N \exp\left(-\epsilon \sum_{t=1}^{T} F^{(t)}\right)$$

$$= Ne^{-\epsilon\mathbb{E}[M]}$$

and therefore

$$(1 - \epsilon)^{m_i} \leq Ne^{-\epsilon\mathbb{E}[M]}$$

$$m_i \ln(1 - \epsilon) \leq -\epsilon\mathbb{E}[M] + \ln N \qquad \text{by taking log on both sides}$$

$$\mathbb{E}[M] \leq \frac{\ln N - m_i \ln(1 - \epsilon)}{\epsilon}$$

$$\mathbb{E}[M] \leq m_i(1 + \epsilon) + \frac{\ln N}{\epsilon} \qquad \text{using a similar approximation from before}$$

which shows that $\mathbb{E}[M] \leq m_i(1 + \epsilon) + O\left(\frac{\ln N}{\epsilon}\right)$. $\qquad \square$

We can see that using our randomised algorithm, we are able to get rid of the factor of 2 that we previously have. Again, if the best expert we have gets the predictions wrong m_i^* times, then $\mathbb{E}[M]$ is bounded by $m_i^*(1 + \epsilon) + O\left(\frac{\ln N}{\epsilon}\right)$.

We can also make a note of a concept called *regret*. The regret is defined as $\mathbb{E}[M] - m_i^*$, which is how many more times you (the aggregator) predicts the wrong outcome compared to the best expert we have. In this algorithm, the regret is bounded by $m_i\epsilon + O\left(\frac{\ln N}{\epsilon}\right)$.