CS 7545: Machine Learning Theory **Fall 2019**

Lecture 6: Weighted Majority Algorithm

Lecturer: Jacob Abernethy *Scribes: Haoliang Jiang and Andi Wang*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

6.1 Review of the last lecture

General setup

- For each round t, every expert i gives a prediction $x_i^t \in \{0, 1\}$;
- An algorithm predicts $\hat{y}^t \in \{0, 1\}$, based on experts' predictions;
- Natural reveals $y^t \in \{0, 1\}$.

Halving algorithm In last lecture, we proved the following claim:

Claim 6.1 *There exists an algorithm which makes fewer than $log_2 N$ mistakes by an arbitrary time $T > 0$,*

$$M_T(\text{Alg}) = \sum_{t=1}^{T} \mathbb{1}[\hat{y^t} \neq y^t] \leq \log_2 N$$

This claimed is proved by considering the *halving algorithm.* Recall that the proof was sketched as follows.

Proof: Let C_{t+1} be the index set of experts who have not yet made mistakes in the first t rounds. We have,

$$C_1 = \{1, 2, 3, ..., N\}$$
$$C_{t+1} = C_t \setminus \{i : x_i^t \neq y^t\}$$
$$\hat{y_t} = \text{round}\left(\frac{1}{|C^t|}\sum_{i \in C_t} x_i^t\right)$$

We can observe that once the algorithm made a mistake, the majority (no less than half) of experts must made a wrong prediction. Then the number of the remaining experts in the next round would be reduced by at least half.

$$\hat{y}^t \neq y^t \implies |C_{t+1}|/|C_t| \leq \frac{1}{2} \implies |C_{t+1}| \leq |C_1|(\frac{1}{2})^{M_t}$$

Suppose the algorithm finds the perfect expert after round T, this means $|C_{T+1}| \leq 1$. Then the maximum M_T satisfies $M_T \leq \log_2 N$, where N is the number of experts in the first round. ■

6.2 The case with multiple choices

Let's now consider a motivating example of betting the winning team of a tournament. Assume that there are n teams. On each round, two teams i_t and j_t play a match. An algorithm aims predicting the outcome of the match: whether i_t beats j_t or vice versa. Assume that each game has only one winner, either i_t or j_t.

Assume that there exists a permutation of all teams $\pi^* \in \mathcal{S}_n$, which specifies the outcome of all matches. (Here $\mathcal{S}_n$ represents the set of all permutations on $[n]$.) Specifically, i_t beats j_t if and only if $\pi(i_t) \geq \pi(j_t)$. Intuitively, there is an absolute ranking of the teams that specifies the outcome of the result. Now, we need an algorithm that minimizes the total number of mistakes.

Claim 6.2 *There exists an algorithm that satisfies the following bound:*

$$C_t \leq \log_2\{n!\}. \leq n \log_2 n$$

where C_t is the smallest number of mistakes that an algorithm makes before time t.

The idea is to consider a committee of $n!$ experts. Each of them corresponds to a permutation $\pi \in \mathcal{S}_n$, and predict the outcome of the game between i_t and j_t according to whether $\pi(i_t) > \pi(j_t)$, i.e.,

$$x_\pi^t = \mathbb{1}[\pi(i_t) > \pi(j_t)].$$

Also, the decision by Nature can be described by permutation π^*,

$$y^t = \mathbb{1}[\pi^*(i_t) > \pi^*(j_t)].$$

Now, we transform the setup of this problem to that of the Halving Algorithm, and that Claim 6.2 follows immediately from Claim 6.1.

Remarks

- The students are encouraged to think about if/how the above algorithm can be performed efficiently.
- By Sterling formula, the expression $\log\{n!\}$ is actually in the order of $n \log n$. Notice that the order is the same with that of the comparisons you need in sort a sequence of size n. However, a major difference is that in sorting problem, we can arbitrarily select the two elements to compare. In our "experts" setup, however, every pair of teams is selected based on the game played, and the measure is the number of mistakes instead of the number of comparisons.
- We might need a smoother algorithm than Halving Algorithm that it might not expire experts once they made a mistake and is similar to a weighted majority vote, as will be introduced in the next section.

6.3 Weighted Majority Algorithm

Now let's consider the case where no expert is "perfect", that is, all experts are subject to making mistakes.

Let $M_T(i) := \sum_{t=1}^T \mathbb{1}\left[x_i^t \neq y^t\right]$ be the number of incorrect guesses from expert i up to time T and $M_T(\text{Alg}) := \sum_{t=1}^T \mathbb{1}\left[\hat{y}^t \neq y^t\right]$ be the total number of incorrect guesses generated from the algorithm that based on the answers of the experts.

We will describe "weighted majority algorithm" (WMA) below, which can be regarded a "softer" or "smoother" version of the Halving algorithm. In WMA, making mistakes will not get an expert "fired". Instead, a mistake downgrade the weight of this expert's guesses. In this sense, the Halving Algorithm is a special case of it, as each expert are assigned with zero or one weight, and making each mistake drops its weight down to zero. It was formally proposed in 1994. However, similar idea appeared around 1950s/1960s.

Weighted Majority Algorithm Let there be N experts in total, and the ith expert make prediction x_i^t at time t. WMA sets weight w_i^t for expert i at each time t, and predicts the output using Algorithm 1 based on the experts' prediction.

The following Theorem 6.3 gives an upper bound on $M_T(\text{WMA})$, the number of wrong decisions $\hat{y}^t$ given by the WMA from time 1 to T.

Theorem 6.3 *For any sequence of data $\vec{x}^1, \ldots, \vec{x}^t$ and $y^1, \ldots, y^t$, we have*

$$M_T(\text{WMA}) \leq \frac{2}{\varepsilon} \log(N) + 2(1+\varepsilon) M_T(i)$$

for any $\varepsilon > 0$ and $i = 1, \ldots, N$, where $\log(\cdot)$ represents the natural logarithm with base e.

Algorithm 1 Weighted Majority Algorithm

```
Initialize $w_i^1 \leftarrow 1, \forall i = 1, \ldots, N$;
for $t = 2, \ldots, T$ do
    Generate prediction at time $t$ by the weighted average of all experts
        $$\hat{y}^t \leftarrow \text{round}\left(\frac{\sum_{i=1}^N w_i^t x_i^t}{\sum_{i=1}^N w_i^t}\right),$$
    Update the weight of all experts with
        $$w_i^{t+1} \leftarrow w_i^t (1 - \varepsilon) \mathbb{1}\left[x_i^t \neq y^t\right].$$
end for
```

Several remarks on the WMA algorithm and Theorem 6.3:

- As the statement in the above theorem holds for every expert, it naturally holds for the "best" expert among all N experts.
- As we select $\varepsilon = 1$, the WMA algorithm reduces to Halving Algorithm and thus also gives an error bound for it. This bound is very similar to the one presented in Lecture 5, only up to a constant factor. In fact, under the setting that there exists an expert i who always makes correct guess, we have $M_T(i) = 0$, and thus $M_T(\text{WMA}) \leq 2\log_e(N)$.
- Consider that you have N machine learning methods to predict the outcome, and you do not know which method performs the best. The theorem here implies that as the number of observations $T \to \infty$, you can always construct an algorithm so that $M_T(\text{Alg})$ increases at the same order as $M_T(i)$, where i represents the algorithm with the best performance. (Note that $2\log_e(N/\varepsilon)$ is neligible as $T \to \infty$). Moreover, this algorithm is constructed according to the WMA.
- The WMA is similar to ensemble learning methods such as *boosting*, though the latter is more complex.

The remaining part of this lecture proves Theorem 6.3. We first need the following lemma:

Lemma 6.4 *The inequalities below hold:*

1. $\log(1 + x) \leq x$ *for all* $x \in \mathbb{R}$;
2. $1 + x \leq \exp(x)$ *for all* $x \in \mathbb{R}$;
3. $\exp(ax) \leq 1 + (e^a - 1)x$ *for all* $x \in [0, 1]$;
4. ***(Challenge Problem)*** : $-\log(1 + x) \leq -x + x^2$ *for all* $x \in [-1/2, 0]$.

Proof:

- The inequality 1 and 2 can be derived from the concavity and convexity of $y = \log(1 + x)$ and $y = \exp\{x\}$, respectively. The tangent line of $y = \log(1 + x)$ at point $x = 0$ is $y = x$, and thus $\log(1 + x) \leq x$. The tangent line of $y = \exp\{x\}$ at point $x = 0$ is $y = x + 1$, and thus $\exp\{x\} \geq x + 1$. These two inequalities are illustrated in the first two plots in Figure 6.1.
- The inequality 3 results from the convexity of function $y = \exp\{ax\}$. As shown in the last plot in Figure 6.1, we have
$$(1 - x)\exp(a \cdot 0) + x\exp(a \cdot 1) \geq \exp\left\{a \cdot [0 \cdot (1 - x) + 1 \cdot x]\right\},$$
which indicates
$$\exp(ax) \leq 1 + (\exp(a) - 1)x.$$

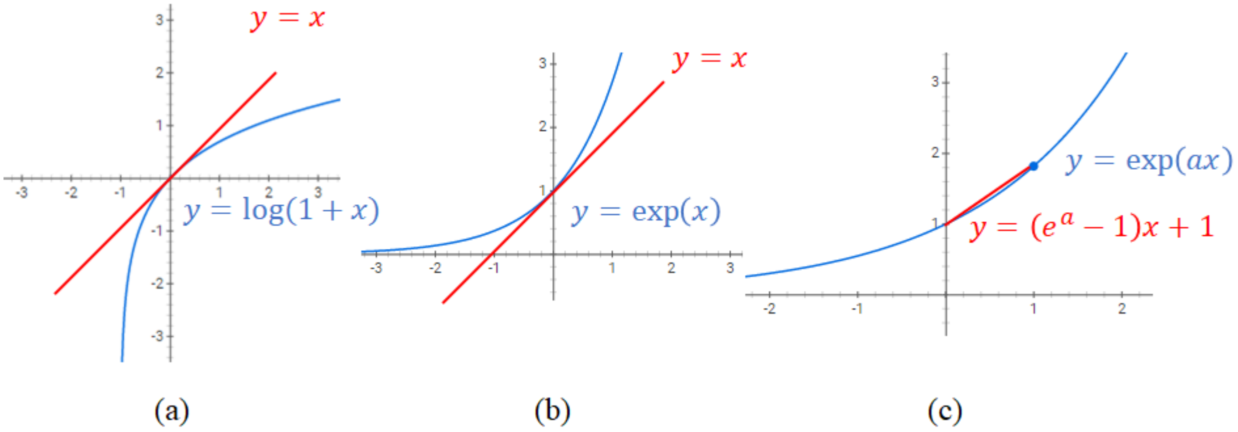

Figure 6.1: Illustrations of the proof of Lemma 6.4

- The inequality 4 is left as an exercise.

■

The Lemma 6.4 will be used to prove Theorem 6.3. The proof also relies on the properties of the potential function, as described below.

Definition 6.5 ***(Potential function)*** *The function $\Phi_t := \sum_{i=1}^N w_i^t$ is called the **potential function**.*

Proposition 6.6 *Based on the WMA algorithm, we have the following facts on the potential function:*

1. *(Initial condition) $\Phi_1 = N$;*
2. *(Lower bound) $\Phi_{T+1} \geq w_i^{T+1} = (1-\varepsilon)^{M_T(i)}$;*
3. *(Upper bound) $\Phi_{T+1} \leq N\left(1-\varepsilon/2\right)^{M_T(\mathrm{WMA})}$.*

Proof:

1. The result follows immediately from the initial condition $w_i^1 = 1, \forall i$.
2. Trivial
3. To prove the upper bound, we need to show

Claim 6.7 *If the WMA errs at time t, then*

$$\Phi_{t+1} \leq (1-\varepsilon/2)\Phi_t.$$

Proof: In class, Prof. Abernethy gave an illustrative interpretation[1]. Based on its idea, a formal proof is provided here. Let $I_t = \{i : x_i^t \neq y^t\}$. We first show that $\sum_{i \in I_t} w_i^t \geq \frac{1}{2}\sum_{i=1}^N w_i^t$. When $y^t = 0$,

[1] Consider N objects in the space whose volumes are $w_i^t, i = 1, \ldots, N$ at time t. They are colored as black and white, and the black objects takes over half of the entire volume. At time $t+1$, the black objects shrink by factor $1-\varepsilon$. Then the total size of the objects shrink by a factor at least $(1-\varepsilon/2)$.

that WMA errs indicates that $\hat{y}^t = 1$, i.e., $I_t = \{i : x_i^t = 1\}$ and $\frac{\sum_{i\in I_t} w_i^t}{\sum_{i=1}^N w_i^t} > 1/2$; when $y^t = 1$, that WMA errs indicates that $\hat{y}^t = 0$, i.e., $I_t = \{i : x_i^t = 0\}$ and $\frac{\sum_{i\in I_t^c} w_i^t}{\sum_{i=1}^N w_i^t} < 1/2$, which further implies $\sum_{i\in I_t} w_i^t \geq \sum_{i=1}^N w_i^t$.

Now

$$\begin{aligned}
\Phi_{t+1} =& \sum_{i=1}^N w_i^{t+1} \\
=& \sum_{i\in I_t} w_i^{t+1} + \sum_{i\in I_t^c} w_i^{t+1} \\
=& \sum_{i\in I_t} (1-\varepsilon) w_i^t + \sum_{i\in I_t^c} w_i^t \\
=& \sum_{i\in I_t} (1-\varepsilon) w_i^t + \left(\sum_{i=1}^N w_i^t - \sum_{i\in I_t} w_i^t\right) \\
=& (-\varepsilon) \sum_{i\in I_t} w_i^t + \sum_{i=1}^N w_i^t \\
\geq& -\frac{\varepsilon}{2} \sum_{i=1}^N w_i^t + \sum_{i=1}^N w_i^t = \left(1 - \frac{\varepsilon}{2}\right) \Phi_t
\end{aligned}$$

■

With the Claim 6.7, we know

$$\begin{aligned}
\Phi_{T+1} &\leq (1-\varepsilon/2)^{\mathbb{1}[y^T \neq \hat{y}^T]} \Phi_T \leq \cdots \\
&\leq \prod_{t=1}^T (1-\varepsilon/2)^{\mathbb{1}[y^t \neq \hat{y}^t]} \Phi_1 = N(1-\varepsilon/2)^{\sum_{t=1}^T \mathbb{1}[\hat{y}^t \neq y^t]} = N(1-\varepsilon/2)^{M_T(\mathrm{WMA})}
\end{aligned}$$

Note that we used Item 1 in Proposition 6.6, $\Phi_1 = N$.

■

Finally, we prove the Theorem 6.3 below by integrating 2 and 3 of Proposition 6.6, as well as Lemma 6.4.

Proof: The upper bound and lower bound of Proposition 6.6 gives

$$(1-\varepsilon)^{M_T(i)} \leq \Phi_{T+1} \leq N(1-\varepsilon/2)^{M_T(\mathrm{WMA})}$$

Take negative-log on both sides, we have

$$-M_T(i)\log(1-\varepsilon) \geq \Phi_{T+1} \geq -M_T(\mathrm{WMA})\log(1-\varepsilon/2) - \log N$$

By Inequality 4 in Lemma 6.4,

$$M_T(i)(\varepsilon+\varepsilon^2) \geq -M_T(i)\log(1-\varepsilon);$$

By Inequality 1 in Lemma 6.4,

$$-M_T(\mathrm{WMA})\log(1-\varepsilon/2) - \log N \geq -\log N + (\varepsilon/2)M_T(\mathrm{WMA}).$$

Combine the above three inequalities, we have

$$M_T(i)(\varepsilon+\varepsilon^2) \geq -\log N + (\varepsilon/2)M_T(\mathrm{WMA})$$

Divide each side of the above inequality by $\varepsilon/2$, we have

$$M_T(WMA) \leq \frac{2}{\varepsilon}\log N + 2(1+\varepsilon)M_T(i).$$

■