Algorithmic Excursions: Topics in Computer Science II **Spring 2016**

Week 9 : Multiplicative Weights Update

Lecturer: Kasturi Varadarajan *Scribe: Vivek Sardeshmukh*

The *multiplicative weights update* is an algorithmic paradigm (such as dynamic programming paradigm and divide & conquer paradigm). A survey by Arora, Hanzon, and Kale [Arora et al., 2012] is an useful resource on this topic. In the first lecture of this week we cover *Weighted Majority Algorithm* and its analysis (Section 9.1). The second lecture of this week will cover *Multiplicative Weighted Algorithm* and its analysis (Section 9.2).

9.1 Weighted Majority Algorithm

Consider the following hypothetical stock scenario: There is a stock that we bought. Every morning the price of the stock either go up or go down. We would like to predict what's going to happen the next day. The stock is hypothetical in a sense that, the up and down in the price of this stock is not based on actual share-market but decided by an adversary. The adversary can see our prediction before deciding what's the status of the stock the next day. That is, an adversary can make us always loose! There is another set of entity involved in this problem: *set of experts*. There are n experts who are also trying to predict the status of the stock for the next day. We can see predictions of all these experts and we want to come up with an algorithm that predicts "as good as" the best experts. Precisely, everyone (experts and us) incur a cost (penalty) of 1 if the prediction for that day is wrong, 0 otherwise. We want to minimize our total cost at the end of day T compared to the best expert. Let cost vector $m_i^{(t)} = 0$ if expert i is correct at time t, 1 otherwise. Similarly define cost vector $m^{(t)} = 0$ if we predicted correctly at time t, 1 otherwise.

That is, if our cost is at time t is $m^{(t)}$ and expert i's cost at time t is $m_i^{(t)}$ then we would like to have minimize the ratio of $\sum_{t=1}^{T} m^{(t)}$ to $\sum_{t=1}^{T} m_i^{(t)}$ for every $i \in [n]$.

To understand the question further, let's try to come up with simple solutions. Then we show that these simple solutions **won't** work motivating a need for sophisticated approach.

Approach 1: Best-Expert-So-Far (wrong approach): Pick the best expert so far (expert whose cost is minimum so far) and predict what this best expert is predicting. To see why this approach is wrong, consider an example where $n = 4$. Let's say initially (on day 1) we picked Expert 1's decision who predicted stock going up. Every other expert predicted stock going down. The adversary made stock to go down incurring a cost of 1 to Expert 1 and us. The best experts so far are expert $\{2, 3, 4\}$. Let's say we picked Expert 2's decision who predicted stock going up. Every other expert predicted stock going down. The adversary made stock to go down incurring a cost of 1 to Expert 2 and us. The similar pattern continues, making each expert $\frac{1}{4}^{th}$ of time wrong and making us all the time wrong. This is not good enough!

Approach 2: Majority-Decision (wrong approach): Pick majority decision from expert's decisions. Note that in this approach we completely ignore the history. This approach also fails. Consider a situation where Expert 1 predicts correctly all the time and all the other experts predicts wrongly. If we follow this approach then we predict wrongly all the time!

What if, we combine the above two approaches, that is, take weighted average of expert's decisions and the weights are calculated based on the history? This is what we are going to describe and analyze formally next.

Algorithm 1 WEIGHTED-MAJORITY

1: Fix $\eta \leq \frac{1}{2}$.
2: For each expert i, set $w_i^{(1)} = 1$.
3: **for** $t \leftarrow 1$ to T **do**
4: Predict according to weighted majority with weights $w_i^{(t)}$
5: Observe cost vectors of each expert $m_i^{(t)}$
6: For each i who predicts wrongly, $w_i^{(t+1)} \leftarrow (1-\eta)w_i^{(t)}$
7: **end for**

Algorithm 1 describes the WEIGHTED-MAJORITY Algorithm. Let

$$M_i^{(t)} = \sum_{t=1}^{T} m_i^{(t)} \text{ and } M^{(T)} = \sum_{t=1}^{T} m^{(t)}.$$

Claim 9.1 *For every expert $i \in [n]$,*

$$M^{(T)} \leq 2(1+\eta)M_i^{(T)} + \frac{2\ln n}{\eta}$$

Proof: Define potential function $\phi^{(t)} = \sum_{i=1}^{n} w_i^{(t)}$. We have, $\phi(1) = n$. If we make a mistake in time t, then in the next steps, weights of at least half of the experts are decreased. Therefore we have,

$$\begin{aligned}
\phi^{(t+1)} &\leq \left(\frac{1}{2} + \frac{1}{2}(1-\eta)\right)\phi^{(t)} \\
&= \left(1 - \frac{\eta}{2}\right)\phi^{(t)} \\
\phi^{(T+1)} &\leq \left(1 - \frac{\eta}{2}\right)^{M^{(T)}} \cdot \phi^{(1)} \\
\phi^{(T+1)} &\leq \left(1 - \frac{\eta}{2}\right)^{M^{(T)}} \cdot n
\end{aligned} \tag{9.1}$$

On the other hand the potential function cannot drop too much. Because, for every i:

$$\begin{aligned}
\phi^{(T+1)} &\geq w_i^{(T+1)} \\
&\geq (1-\eta)^{M_i^{(T)}}
\end{aligned} \tag{9.2}$$

From Equations 9.1 and 9.2 we have,

$$\begin{aligned}
(1-\eta)^{M_i^{(T)}} &\leq \left(1 - \frac{\eta}{2}\right)^{M^{(T)}} \cdot n \\
&\leq n \cdot \exp\left(-M^{(T)}\frac{\eta}{2}\right)
\end{aligned}$$

$$
\begin{aligned}
\exp\left(\frac{\eta M^{(T)}}{2}\right) &\le n \cdot \left(\frac{1}{1-\eta}\right)^{M_i^{(T)}} \\
\frac{\eta M^{(T)}}{2} &\le \ln n + M_i^{(T)} \cdot \ln \frac{1}{1-\eta} \\
&\le \ln n + M_i^{(T)} \cdot (\eta + \eta^2) \qquad (9.3) \\
M^{(T)} &\le \frac{2 \ln n}{\eta} + 2(1+\eta) M_i^{(T)}
\end{aligned}
$$

Equation 9.3 follows from the following identity:

$$\ln \frac{1}{1-x} \le x + x^2 \qquad (9.4)$$

■

9.2 Multiplicative Weights Algorithm

In the earlier setting, we only accounted for "loss". That is the entities involved (experts and us) incur a cost of 0 or +1. In this section, we'll consider a more complex setting, where for each correct prediction entities get benefit and for each wrong prediction incur a cost. We first state this setting formally and design an algorithm.

There are n experts, each offering a suggestion at each time instant. At each time instant t:

- Our algorithm needs to pick one of the decisions
- cost/loss vector $m^{(t)} \in [-1, +1]$ is revealed

The algorithm should minimize the overall cost at the end of time T (with respect to the "best" expert). We devise a randomized algorithm to solve this problem (Multiplicative Weight (MW Algorithm)). At a high level, the MW Algorithm increases weights if there is a profit at that time instant else decreases. It picks an expert randomly where the probability depends on weights. See Algorithm 2 for precise statements.

Algorithm 2 MW Algorithm

Fix $\eta \le \frac{1}{2}$.
Set $w_i^{(1)} = 1$ for each i.
for $t \leftarrow 1$ to T **do**
    Choose decision i with probability proportional to $w_i^{(t)}$. That is, use probability distribution

$$p^{(t)} = \left\{ \frac{w_1^{(t)}}{\phi^{(t)}}, \frac{w_2^{(t)}}{\phi^{(t)}}, \ldots, \frac{w_n^{(t)}}{\phi^{(t)}} \right\}$$

    Observe cost vector $m^{(t)}$
    Set $w_i^{(t+1)} = w_i^{(t)} \left(1 - \eta m_i^{(t)}\right)$
end for

The expected cost of the algorithm at time t is

$$p^{(t)}m^{(t)} = \sum_{i=1}^{n} p_i^{(t)} m_i^{(t)}.$$

We prove the following guarantee on the expected cost of the above algorithm:

Claim 9.2 *For any expert $i \in [n]$,*

$$\sum_{t=1}^{T} p^{(t)}m^{(t)} \leq \sum_{t=1}^{T} m_i^{(t)} + \eta \sum_{t=1}^{T} \left| m_i^{(t)} \right| + \frac{\ln n}{\eta}.$$

Proof: We use the potential function $\phi^{(t)} = \sum_i w_i^{(t)}$ to prove this claim.

$$\begin{aligned}
\phi^{(t+1)} &= \sum_i w_i^{(t+1)} \\
&= \sum_i w_i^{(t)} \left(1 - \eta m_i^{(t)}\right) \\
&= \phi^{(t)} - \eta \phi^{(t)} \sum_i \left(m_i^{(t)} p_i^{(t)}\right) \\
&= \phi^{(t)} \left(1 - \eta m^{(t)} p^{(t)}\right) \\
&\leq \phi^{(t)} \exp\left(-\eta m^{(t)} p^{(t)}\right)
\end{aligned}$$

$$\phi^{(T+1)} \leq \phi^{(1)} \exp\left(-\eta \sum_{t=1}^{T} m^{(t)} p^{(t)}\right) \tag{9.5}$$

On the other hand, the potential function cannot drop too much:

$$\begin{aligned}
\phi^{(T+1)} &\geq w_i^{(T+1)} \\
&= \prod_{t=1} T \left(1 - \eta m_i^{(t)}\right) \\
&\geq \prod_{t: m_i^{(t)} \geq 0} (1-\eta)^{m_i^{(t)}} \prod_{t: m_i^{(t)} < 0} (1+\eta)^{-m_i^{(t)}} && (9.6) \\
&\geq (1-\eta)^{\sum_{\geq 0} m_i^{(t)}} \cdot (1+\eta)^{-\sum_{<0} m_i^{(t)}} && (9.7)
\end{aligned}$$

Equation 9.6 follows from using the following inequalities:

$$\begin{aligned}
(1-\eta)^x &\leq (1-\eta x) && \text{if } x \in [0,1] \\
(1+\eta)^{-x} &\leq (1+\eta x) && \text{if } x \in [-1,0]
\end{aligned}$$

From Equation 9.5 and 9.7 and taking logarithm we have,

$$\begin{aligned}
\ln n - \eta \sum_{t=1}^{T} \left(m^{(t)} p^{(t)} \right) &\geq \sum_{\geq 0} \left(m_i^{(t)} \ln(1-\eta) \right) - \sum_{<0} \left(m_i^{(t)} \ln(1+\eta) \right) \\
\sum_{t=1}^{T} \left(m^{(t)} p^{(t)} \right) &\leq \frac{\ln n}{\eta} - \frac{1}{\eta} \sum_{\geq 0} \left(m_i^{(t)} \ln(1-\eta) \right) + \frac{1}{\eta} \sum_{<0} \left(m_i^{(t)} \ln(1+\eta) \right) \\
&\leq \frac{\ln n}{\eta} + \frac{1}{\eta} \sum_{\geq 0} \left(m_i^{(t)} \left(\eta + \eta^2\right) \right) + \frac{1}{\eta} \sum_{<0} \left(m_i^{(t)} \left(\eta - \eta^2\right) \right) \qquad (9.8) \\
&\leq \frac{\ln n}{\eta} + \sum_{t=1}^{T} m_i^{(t)} + \eta \sum_{t=1}^{T} \left| m_i^{(t)} \right|
\end{aligned}$$

Equation 9.8 follows from Equation 9.4 and the following identity:

$$\ln(1+x) \geq x - x^2$$

■

The following corollary follows. This corollary is useful in various applications that we'll see in the next week.

Corollary 9.3 *For any probability vector $q \in \mathbb{R}^n$,*

$$\sum_{t=1}^{T} m^{(t)} p^{(t)} \leq \left(\sum_{i=1}^{T} m^{(t)} + \eta \left| m^{(t)} \right| \right) \cdot q + \frac{\ln n}{n}$$

where $\left| m^{(t)} \right| = (\lambda, \lambda, \ldots, \lambda)$ is a n-length vector and $\lambda = \max_i \left| m_i^{(t)} \right|$.

References

[Arora et al., 2012] Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164.