Multiplicative Weights Algorithm

1 Motivation

Multiplicative weights is an algorithm in online learning which combines everything we've learned so far. Beyond online learning, it can be viewed much more broadly for classification, LP solving, gameplay, convex programming, and more. It's sort of a proto-gradient descent algorithm and is a basic form of machine learning.

The most basic setup is that we have a binary decision to make in each timestep (buy or sell, etc.). After the algorithm commits to a decision, nature decides the outcome. The algorithm's goal is to minimize the number of mistakes, or maximize the number of correct decisions.

In a pure online algorithms setting (competitive analysis), there is no hope for this algorithm because the offline OPT would get everything right, and the online gets everything wrong. So, we define a new measure of success. We'll compare our algorithm to n "experts" (note that they are not necessarily domain experts, just benchmarks designed by an adversary). During each timestep t, each expert i makes a recommendation to the algorithm. At the end of the process, the algorithm is evaluated by how well it did compared to the best expert in hindsight.

2 Weighted Majority Algorithm

A basic method to solve this problem is to make a decision based on a majority of experts, weighted in regards to how many mistakes they've made so far.

Algorithm 1 Weighted Majority Algorithm

Fix a parameter $\eta \leq \frac{1}{2}$ (usually 0.01)
Initialize all $w_i^{(1)} = 1$
for each timestep t: **do**
    Choose an action based on the weighted majority of experts with weights $w_i^{(t)}$.
    for every correct expert: **do**
        Set $w_i^{(t+1)} = w_i^{(t)}$
    end for
    for every incorrect expert: **do**
        Set $w_i^{(t+1)} = (1-\eta)w_i^{(t)}$
    end for
end for

Let $M^{(T)}$ be the number of mistakes that weighted majority makes up to time T, and $m_i^{(T)}$ be the number of mistakes that expert i makes up to time T. And, let the total weight $\Phi^{(t)} := \sum_i w_i^{(t)}$.

Theorem 1. $M^{(T)} \leq \frac{2}{1-\eta} m_i^{(T)} + \frac{2}{\eta}\ln(n)$, *with the tightest bound achieved by letting i be the best expert.*

Proof. Whenever the algorithm makes a mistake, $\leq \frac{1}{2}$ of the expert weight gets multiplied by $(1-\eta)$. Thus:

$$\Phi^{(t+1)} \leq \frac{1}{2}\Phi^{(t)} + \frac{1}{2}\Phi^{(t)}(1-\eta)$$

$$= \Phi^{(t)}(1-\frac{\eta}{2})$$

$\Phi^{(1)} = n$, so by induction:

$$\Phi^{(t)} \leq n(1-\frac{\eta}{2})^{M^{(t)}}$$

In other words, the overall weight in the system drops exponentially in the number of our algorithm's mistakes. For any expert i,

$$w_i^{(T)} = (1-\eta)^{m_i^{(t)}}$$

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

$$(1-\eta)^{m_i^{(T)}} \leq n(1-\frac{\eta}{2})^{M^{(T)}}$$

$$(\frac{2}{2-\eta})^{M^{(T)}} \leq n(1-\frac{\eta}{2})^{m_i^{(T)}}$$

$$M^{(T)} \leq m_i^{(T)} \frac{\log(1+\frac{\eta}{1+\eta})}{\log(1+\frac{\eta}{2-\eta})} + \frac{\log(n)}{\log(1+\frac{\eta}{2+\eta})}$$

A useful trick here is that, by the Taylor expansion, $x - \frac{x^2}{2} \leq \ln(1+x) \leq x$. So:

$$\ln(1+\frac{\eta}{2-\eta}) \geq \frac{\eta}{2-\eta} - \frac{1}{2}(\frac{\eta}{2-\eta})^2$$

$$= \frac{4\eta - 3\eta^2}{2(2-\eta)^2}$$

$$\geq \frac{\eta(4-4\eta+\eta^2)}{2(2-\eta)^2}$$

$$= \frac{\eta}{2}$$

And:

$$\ln(1+\frac{\eta}{1-\eta}) \leq \frac{\eta}{1-\eta}$$

So the messy $M^{(T)}$ equation above becomes:

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

$$= \frac{2}{1-\eta} m_i^{(T)} + \frac{2}{\eta}\ln(n)$$

$\square$

3 Multiplicative Weights Algorithm

Weighted majority is pretty good, but the binary decision restricts the scenarios we can apply it to. In a more general case, imagine that there are a wide variety of options each expert can select, and each have an associated reward/penalty. For example, instead of choosing to buy or sell some stock, the expert has to recommend a stock to invest in.

Formally, in each round t, we will pick one expert to copy. Then, each expert i receives a penalty $m_i^{(t)} \in [-1, 1]$. We receive the penalty of the expert we copied. Then, we learn all penalties of all experts for the round, and repeat. This is similar to Multi-Armed Bandit, but in that scenario we learn only the payoff of the expert we copied, so there is a complex balance between exploration and exploitation.

Our key new insight is that instead of choosing deterministically, which is easy for an adversary to exploit, we can randomize between experts.

Algorithm 2 Multiplicative Weights Algorithm

```
Fix a parameter η ≤ 1/2 (usually 0.01)
Initialize all w_i^(1) = 1
for each timestep t: do
    Let Φ(t) = Σ_i w_i^(t)
    Choose an expert i with probability p_i^(t) = w_i^(t) / Φ^(t)
    for each expert i: do
        Set w_i^(t+1) = w_i^(t) * (1 − η)m_i^(t)
    end for
end for
```

Let $M^{(T)}$ represent the total penalty of multiplicative weights up to time T, and $m_i^{(T)}$ be the total penalty of expert i up to time T.

Theorem 2. *$M^{(T)} \leq m_i^{(T)} + \frac{\eta}{1-\eta}|m_i^{(T)}| + \frac{1}{\eta}\ln(n)$, with the tightest bound achieved by letting i be the best expert.*

Proof. Similarly to weighted majority, we want to compare the sum of all weights to the sum of a single expert.

$$\Phi^{(t+1)} = \sum_i w_i^{(t+1)}$$

$$= \sum_i (w_i^{(t)} * (1-\eta)m_i^{(t)})$$

$$= \Phi^{(t)} - \eta \sum_i (\frac{w_i^{(t)}}{\Phi^{(t)}} \Phi^{(t)} m_i^{(t)})$$

$$= \Phi^{(t)}(1 - \eta \sum_i (p_i^{(t)} m_i^{(t)}))$$

$$= \Phi^{(t)}(1 - \eta \vec{p}^{(t)} \cdot \vec{m}^{(t)})$$

Because $(1 + x) \leq e^x$:

$$\leq \Phi^{(t)} \exp(-\eta \vec{p}^{(t)} \cdot \vec{m}^{(t)})$$

By unrolling,

$$\Phi^{(t+1)} \leq \Phi^{(1)} * \prod_{t \leq T} \exp(-\eta \vec{p}^{(t)} \cdot \vec{m}^{(t)})$$

$$= n \exp(-\eta M^{(T)})$$

For any expert i:

$$w_i^{(t+1)} = \prod_{t \leq T} (1 - \eta m_i^{(t)})$$

$$= \prod_{t \leq T, m_i^{(t)} \geq 0} (1 - \eta m_i^{(t)}) * \prod_{t \leq T, m_i^{(t)} < 0} (1 - \eta m_i^{(t)})$$

Because $m_i^{(t)} \in [-1, 1]$:

$$\geq \prod_{t \leq T, m_i^{(t)} \geq 0} (1 - \eta)^{m_i^{(t)}} * \prod_{t \leq T, m_i^{(t)} < 0} (1 + \eta)^{-m_i^{(t)}}$$

$$= (1 - \eta)^{\sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)}} * (1 + \eta)^{-\sum_{t \leq T, m_i^{(t)} < 0} m_i^{(t)}}$$

As before, $\Phi^{(t+1)} \geq w_i^{(t+1)}\ \forall i$, so:

$$n \exp(-\eta M^{(T)}) \geq (1 - \eta)^{\sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)}} * (1 + \eta)^{-\sum_{t \leq T, m_i^{(t)} < 0} m_i^{(t)}}$$

$$M^{(T)} \leq \frac{1}{\eta}(\ln(n) + \ln(\frac{1}{1 - \eta}) \sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)} + \ln(1 + \eta) \sum_{t \leq T, m_i^{(t)} < 0} m_i^{(t)})$$

By Taylor expansion bounds, $x - \frac{x^2}{2} \leq \ln(1 + x) \leq x$, so:

$$M^{(T)} \leq \frac{1}{\eta}(\ln(n) + \frac{\eta}{1 - \eta} \sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)} + (\eta - \frac{\eta^2}{2}) \sum_{t \leq T, m_i^{(t)} < 0} m_i^{(t)})$$

$$= \frac{1}{1 - \eta} \sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)} + (1 - \frac{\eta}{2}) \sum_{t \leq T, m_i^{(t)} < 0} m_i^{(t)} + \frac{1}{\eta} \ln(n)$$

$$= \sum_{t \leq T} m_i^{(t)} + (1 - \frac{\eta}{2}) + \frac{\eta}{1 - \eta} \sum_{t \leq T, m_i^{(t)} \geq 0} m_i^{(t)} + \frac{\eta}{2} \sum_{t \leq T, m_i^{(t)} < 0} |m_i^{(t)}| + \frac{1}{\eta} \ln(n)$$

$$\leq m_i^{(T)} + \frac{\eta}{1 - \eta} |m_i^{(T)}| + \frac{1}{\eta} \ln(n)$$

$\square$

Corollary 3. *For the goal of maximizing rewards, multiplicative weights guarantees that our reward up to time T is $R^{(T)} \leq m_i^{(T)} + \frac{\eta}{1-\eta} |m_i^{(T)}| + \frac{1}{\eta} \ln(n)$.*

Proof. Flip all the signs in the previous proof. □

Corollary 4. *If $\vec{p}$ is any distribution over experts, the cost is at most $\sum_t (\vec{m}^{(t)} + \eta|\vec{m}^{(t)}|) \cdot \vec{p} + \frac{1}{\eta}\ln(n)$. Conceptually, this means that if you're doing as well as all the experts, you're also doing as well as any mixture.*

Proof. Add the inequalities for all i scaled by p_i. □

4 Multiplicative weights for classification

Given m points in an n-dimensional space, where point $\vec{a_j}$ has label $l_j \in \{-1, 1\}$. We want to learn non-negative weights x_i on dimensions i such that on all points j:

$$\mathrm{sgn}(\vec{a_j} \cdot \vec{x}) = l_j$$

All positive examples have $\vec{a_j} \cdot \vec{x} > 0$, while all negative examples have $\vec{a_j} \cdot \vec{x} < 0$. By dividing each $\vec{a_j}$ by l_j, we standardize our goal to be $\vec{a_j} \cdot \vec{x} > 0 \ \forall j$. From there, we scale all $a_{j,i}$ by $\max_{j,i} |a_{j,i}|$, so without loss of generality, all $a_{j,i} \in [-1, 1]$. Finally, we scale $\vec{x}$ by $\sum_i x_i$, so without loss of generality $\sum_i x_i = 1$.

Through these preprocessing steps, our new goal becomes:

Find $\vec{x}$ satisfying:

$$\begin{cases} \vec{a_j} \cdot \vec{x} > 0 & \forall j \\ \sum_i x_i = 1 & \\ x_i \geq 0 & \forall i \end{cases}$$

Which is an LP! Like support vector machines from machine learning, we assume a margin $\mathcal{E} \leq \frac{1}{2}$ for classification because LPs cannot handle strict inequalities. This means that there is a vector $\vec{x}^*$ such that $\vec{a_j} \cdot \vec{x}^* \geq \mathcal{E} \ \forall j$.

We want to solve this LP with multiplicative weights. Our approach is that each variable i is an expert, and x_i is our weight on expert i. Then, we run multiplicative weights for reward maximization with $\eta = \frac{\mathcal{E}}{2}$.

If we find $\vec{x}$ satisfying the LP, we are done. Until then, in each round t, there must be some $j^{(t)}$ with $\vec{a_j} \cdot \vec{x}^{(t)} \leq 0$. For each t, we'll make the expert payoff vector $a_{j^{(t)}} := m^{(t)}$. Then, our reward in each round ≤ 0 and x^*gets at least $\mathcal{E}$. We can now apply Corollary 4:

$$0 \geq R^{(T)}$$

$$\geq \sum_t (\vec{a_j}^{(t)} \cdot \vec{x}^*) - \frac{\eta}{1-\eta} \sum_t (|\vec{a_j}^{(t)}| \cdot \vec{x}^*) - \frac{1}{\eta}\ln(n)$$

$$\geq \mathcal{E}T - \frac{\frac{\mathcal{E}}{2}}{1 - \frac{\mathcal{E}}{2}} T - \frac{2}{\mathcal{E}}\ln(n)$$

Because $\mathcal{E} \leq \frac{1}{2}$:

$$\geq \frac{\mathcal{E}}{3} T - \frac{2}{\mathcal{E}}\ln(n)$$

Thus:

$$T \leq \frac{6}{\mathcal{E}^2} \ln(n)$$

So we are guaranteed to find our classification vector in no more than $\frac{6}{\mathcal{E}^2} \ln(n)$ iterations of multiplicative weights.