

Chapter 13

Multiplicative Weight Update I: Learning from Experts

We will cover a new algorithmic technique in this part of the book, called the **Multiplicative Weight Update (MWU)** technique. You should really think of MWU as something like *dynamic programming* or *greedy* approaches in algorithm design; it is not a single algorithm but rather a general technique for designing algorithms for various problems and scenarios. We refer the interested reader to the survey [AHK12] for (a lot) more details on MWU and its origin.

13.1 The Experts Problem

We start by looking at this technique for an abstract problem in online learning, sometimes called *the experts problem* or the problem of *learning from experts*. This will help motivate one of the uses of the MWU technique.

Problem 13.1. Suppose we have a set of n “experts” who make predictions about a certain outcome each day (say, whether it is going to rain today or not). Abstractly, think of the *universe* of possible options for each day to be a set U . At the beginning of each day t :

1. Every expert $i \in [n]$ makes a **prediction** $p_i^{(t)} \in U$. Then, we, as the algorithm designer, receive all these predictions and have to come up with our **own prediction** $o^{(t)} \in U$ for that day (say, we predict it will rain that day).
2. After we made our prediction for that day, we get to see the **actual outcome** $a^{(t)} \in U$ of that day (say, whether or not it actually rained).

The goal in this problem is to minimize the number of **mistakes** we made, namely, the number of days that our own prediction was different from the actual outcome.

Notice that our only access to the “ground truth” of the universe is through these experts. This means that we cannot expect to minimize the number of mistakes in an *absolute* sense: if the experts are also making a lot of mistakes, there is no hope for us to not do the same (specifically, think of the universe U to be unbounded or even infinite, so it is not like we can just “guess” the right answer randomly at each point). So, it makes sense to compare our number of mistakes with respect to the number of mistakes made by these experts.

It turns out that the *right* choice here is to compare ourselves with that of the *best expert* in

hindsight; i.e., after the T days of the game are finished, we would like to say that our number of mistakes is not much larger than that of the expert with the minimum mistakes in these T days. This allows us to minimize our “regret” in not having listened to this particular expert throughout. Thus, our goal is to minimize the number of mistakes we have to do compared to the number of mistakes of the best expert. Specifically, let m denote the number of mistakes by our algorithm, and m_i denote the mistakes of the expert $i \in [n]$. Our goal is for m to be close to $m^* = \min_{i \in [n]} m_i$.

13.1.1 Warm-Up 1: An *Ideal* Expert

Let us first consider an easy case of this problem when $m^* = 0$, i.e., there is an expert that does not make any mistake. Here, there is a very simple algorithm:

Algorithm 13.2.

1. Let N be the set of all experts initially.
2. Every day, return the *majority* prediction of all experts in N as your own prediction (breaking the ties arbitrarily).
3. Upon seeing the actual outcome, remove any expert from N with wrong prediction.

Lemma 13.3. *Algorithm 13.2 makes $\lceil \log n \rceil$ mistakes in total.*

Proof. Let $N^{(t)}$ denote the set of all experts still present at the beginning of a day $t \geq 1$. Suppose we make a mistake at day t ; then, we know that at least half the experts in $N^{(t)}$ also made a mistake (because we went with the majority prediction that turns out to be wrong). Thus, we will have

$$|N^{(t+1)}| \leq \frac{1}{2} \cdot |N^{(t)}|.$$

In other words, for any mistake Algorithm 13.2 makes, the number of remaining experts is at least halved. At the same time, by our assumption that $m^* = 0$, there is an expert that is never removed from N . Thus, $|N^{(t)}| \geq 1$ for all $t \geq 1$. This means that the total number of mistakes we can have is at most $\lceil \log n \rceil$ as otherwise we will run out of experts. $\square$

13.1.2 Warm-Up 2: A Crude Approximation

Suppose now we do not have the assumption that $m^* = 0$, i.e., we are in the general case. Let us start with a crude approximation algorithm that is a direct generalization of Algorithm 13.2.

Algorithm 13.4.

1. Run Algorithm 13.2 and remove any expert with a mistake from N .
2. Once N becomes empty, bring back all the experts to N and repeat from Line (1).

Lemma 13.5. *Algorithm 13.4 makes $m^* \cdot \lceil \log n \rceil + \lceil \log n \rceil$ mistakes in total.*

Proof. Let i^* be the index of the best expert so $m_{i^*} = m^*$. Consider one *epoch* as the consecutive days when we start with N equal to all experts until it becomes completely empty and we restart. By the same argument as in Lemma 13.3, during an epoch, the total number of mistakes we have is $\lceil \log n \rceil$. On the other hand, the total number of epochs is at most m^* because each time N becomes empty, it means i^* made a mistake, but the total mistakes of i^* is m^* . Finally, after all the epochs, there can be at most another $\lceil \log n \rceil$ more mistakes (and this time N will not become empty) by Lemma 13.3 again. This proves the lemma. $\square$

13.2 The Weighted Majority Algorithm

Let us now consider the first “real” algorithm for the problem. It is clear that [Algorithm 13.4](#) is quite “aggressive”: it removes an expert based on just one mistake, and then brings them back at a later time without even taking into account the mistakes made by the expert during this time. We can instead consider a more “smooth” version of this algorithm, called the **weighted majority algorithm**: we assign weights to the predictions of the experts and whichever expert makes a mistake will get a lower weight; then, when it comes to predicting the outcome, we work with the weighted majority of the experts (hence, the name); [Figure 13.1](#) illustrates one round of this weight update.

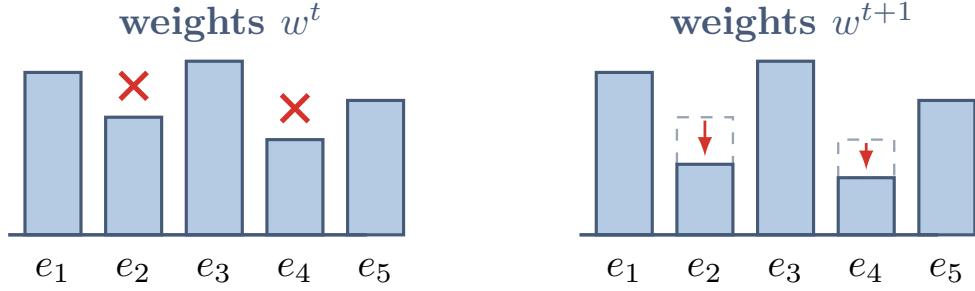


Figure 13.1: One round of the weighted majority update ([Algorithm 13.6](#)): each expert $i \in [n]$ has a weight $w_i^{(t)}$ for each day $t \in [T]$; the algorithm predicts by weighted majority, and every expert wrong that day has its weight multiplied by $(1 - \eta)$, which shrinks its bar (here, the experts marked by red $\times$ have their weight reduced).

Algorithm 13.6.

1. For every expert $i \in [n]$, maintain a weight $w_i^{(t)}$ for every day t . Initially, $w_i^{(1)} = 1$ for all $i \in [n]$.
2. Every day $t \geq 1$, return the weighted majority of the predictions of all experts in $[n]$, based on their weights in $w^{(t)}$, as your own prediction (breaking the ties arbitrarily).
3. Upon seeing the outcome, update the weight of any wrong expert $i \in [n]$ to be $w_i^{(t+1)} = (1 - \eta) \cdot w_i^{(t)}$ for some parameter $\eta \in (0, 1)$ that we can choose later.

Lemma 13.7. [Algorithm 13.6](#) makes at most $2 \cdot (1 + \eta) m^* + \frac{2 \ln n}{\eta}$ mistakes for any $\eta \in (0, 1/2)$.

Proof. Define the following *potential function* for every $t \geq 1$,

$$W^{(t)} := \sum_{i=1}^n w_i^{(t)}.$$

Consider any day $t \geq 1$ that we make a mistake in. We have,

$$\begin{aligned} W^{(t+1)} &= \sum_{i: \text{correct}} w_i^{(t+1)} + \sum_{i: \text{wrong}} w_i^{(t+1)} && \text{(by the definition of } W^{(t+1)}) \\ &= \sum_{i: \text{correct}} w_i^{(t)} + \sum_{i: \text{wrong}} (1 - \eta) \cdot w_i^{(t)} && \text{(by the definition of the update rule)} \\ &= W^{(t)} - \eta \cdot \sum_{i: \text{wrong}} w_i^{(t)} && \text{(by the definition of } W^{(t)}) \end{aligned}$$

$$\begin{aligned}
&\leq W^{(t)} - \frac{\eta}{2} \cdot W^{(t)} \\
&\quad \text{(because if we made a mistake, the weighted majority also made a mistake)} \\
&= \left(1 - \frac{\eta}{2}\right) \cdot W^{(t)}.
\end{aligned} \tag{13.1}$$

Let m denote the total number of mistakes we made till the last day T . Then,

$$W^{(T)} \leq \left(1 - \frac{\eta}{2}\right)^m \cdot W^{(1)} = \left(1 - \frac{\eta}{2}\right)^m \cdot n,$$

where we used [Eq \(13.1\)](#) repeatedly for the first inequality, and the fact that for all $i \in [n]$, $w_i^{(1)} = 1$ for the equality. This means that as long as we make “many” mistakes, the value of our potential function drops “a lot” also. This is similar to the first part of the proof of [Lemma 13.3](#) that argued that each mistake cuts the number of experts—which can be thought of as the potential function in that lemma—by half.

On the other hand, we argue that the potential function cannot drop by a lot either. Let i^* be the index of the best expert so $m_{i^*} = m^*$. We have

$$w_{i^*}^{(T+1)} = (1 - \eta)^{m^*},$$

because we only reduce the weight of i^* if it makes a mistake, and it can only make a mistake m^* times. But since $w_{i^*}^{(T+1)} \leq W^{(T+1)}$, we have,

$$(1 - \eta)^{m^*} = w_{i^*}^{(T+1)} \leq W^{(T+1)} \leq \left(1 - \frac{\eta}{2}\right)^m \cdot n.$$

By taking the natural log of both sides, we have,

$$m^* \cdot \ln(1 - \eta) \leq m \cdot \ln\left(1 - \frac{\eta}{2}\right) + \ln n. \tag{13.2}$$

We now use the following two inequalities.

Fact 13.8. For any $x \in (0, 1/2)$,

$$-(x + x^2) \leq \ln(1 - x) \leq -x.$$

Proof. Follows from the Taylor expansion of $-\ln(1 - x) = x + x^2/2 + x^3/3 + x^4/4 + \dots$. $\square$

Applying [Fact 13.8](#) to [Eq \(13.2\)](#), we get

$$-m^* \cdot (\eta + \eta^2) \leq -m \cdot \frac{\eta}{2} + \ln n,$$

which, by reorganizing terms, implies that

$$m \leq 2 \cdot (1 + \eta) \cdot m^* + \frac{2 \ln n}{\eta},$$

as desired. $\square$

[Lemma 13.7](#) says that we can get much closer to the mistakes of the best expert, almost by a factor 2, at the cost of paying some extra additive factors of $\ln n$. Can we avoid this factor 2? The answer is *No*.

Proposition 13.9. Any deterministic algorithm for [Problem 13.1](#) makes at least $2m^*$ mistakes on some inputs.

Proof. Consider an instance with two experts, one of them always predicting 0 and the other, always predicting 1. Now, since the algorithm is deterministic, at any given day, we can make sure the output prediction of the algorithm is wrong. But, since there are two experts and with binary answers, at least one of the experts can only make an error half the days. So, we make at least twice as many mistakes as the best expert, concluding the proof. $\square$

Remark. In all the algorithms we analyzed so far, we could have actually “penalized” the experts *only* on the days when our prediction was wrong, not on every day that the expert was wrong. It is easy to see that our analysis never took into account the mistakes by the experts on other days. But, that seems inefficient although [Proposition 13.9](#) suggests that we could not have improved this much further. We show how to exploit this using *randomization* in the next algorithm.

13.3 The Multiplicative Weight Update Algorithm

Let us now present the final algorithm which is *randomized* and is often considered as ‘the’ multiplicative weight update (MWU) algorithm (even though all other algorithms we saw also can be called MWU technically). We are going to use randomization in a very simple way to bypass the limit posed in [Proposition 13.9](#) (basically, and for the most part, to break the ties).

Algorithm 13.10.

1. For every expert $i \in [n]$, maintain a weight $w_i^{(t)}$ for every day t . Initially, $w_i^{(1)} = 1$ for all $i \in [n]$.
2. Every day $t \geq 1$, pick one expert $i \in [n]$ randomly such that the probability that expert $i \in [n]$ is chosen is proportional to its weight, i.e., is

$$\frac{w_i^{(t)}}{\sum_{j=1}^n w_j^{(t)}};$$

then return the prediction of this expert as your own prediction.

3. Upon seeing the outcome, update the weight of any wrong expert $i \in [n]$ to be $w_i^{(t+1)} = (1 - \eta) \cdot w_i^{(t)}$ for some parameter $\eta \in (0, 1)$ that we can choose later.

Notice that the only difference between [Algorithm 13.10](#) and our previous [Algorithm 13.6](#) is in how we output our own prediction – the update rule is exactly as before.

Lemma 13.11. *Algorithm 13.10 makes at most $(1 + \eta) m^* + \frac{\ln n}{\eta}$ mistakes in expectation for any $\eta \in (0, 1/2)$.*

Proof. As in [Lemma 13.7](#), define the following potential function for every $t \geq 1$,

$$W^{(t)} := \sum_{i=1}^n w_i^{(t)}.$$

Moreover, for every $t \geq 1$, define

$$B^{(t)} := \frac{1}{W^{(t)}} \cdot \sum_{i: \text{wrong}} w_i^{(t)}.$$

Notice that at any day $t \geq 1$,

$$\Pr(\text{output prediction is wrong}) = B^{(t)},$$

because we are picking the experts at day t according to their weight $w^{(t)}$. As such, letting m denote the number of mistakes in all the T days in our algorithm, we have,

$$\mathbb{E}[m] = \sum_{t=1}^T B^{(t)},$$

by linearity of expectation.

On the other hand, for *every* day $t \geq 1$ (and not only the ones we make a mistake in),

$$\begin{aligned} W^{(t+1)} &= \sum_{i: \text{correct}} w_i^{(t+1)} + \sum_{i: \text{wrong}} w_i^{(t+1)} && \text{(by the definition of } W^{(t+1)}) \\ &= \sum_{i: \text{correct}} w_i^{(t)} + \sum_{i: \text{wrong}} (1 - \eta) \cdot w_i^{(t)} && \text{(by the definition of the update rule)} \\ &= W^{(t)} - \eta \cdot \sum_{i: \text{wrong}} w_i^{(t)} && \text{(by the definition of } W^{(t)}) \\ &\leq W^{(t)} - \eta \cdot B^{(t)} \cdot W^{(t)} && \text{(by the definition of } B^{(t)}) \\ &= (1 - \eta \cdot B^{(t)}) \cdot W^{(t)}. \end{aligned}$$

Let i^* be the index of the best expert so $m_{i^*} = m^*$. As before,

$$w_{i^*}^{(T+1)} = (1 - \eta)^{m^*},$$

because we only reduce the weight of i^* if it makes a mistake, and it can only make a mistake m^* times. But since $w_{i^*}^{(T+1)} \leq W^{(T+1)}$, we have,

$$\begin{aligned} (1 - \eta)^{m^*} = w_{i^*}^{(T+1)} &\leq W^{(T+1)} \leq \prod_{t=1}^T (1 - \eta \cdot B^{(t)}) \cdot n \\ &\leq \exp\left(-\sum_{t=1}^T \eta \cdot B^{(t)} + \ln n\right) = \exp(-\eta \cdot \mathbb{E}[m] + \ln n), \end{aligned}$$

where we used the inequality $1 - x \leq e^{-x}$ for $x \in (0, 1)$.

By taking the natural log of both sides, we have,

$$m^* \cdot \ln(1 - \eta) \leq -\eta \cdot \mathbb{E}[m] + \ln n.$$

And using the inequality $\ln(1 - x) \geq -(x + x^2)$ (by [Fact 13.8](#)), we get,

$$-m^* \cdot (\eta + \eta^2) \leq -\eta \cdot \mathbb{E}[m] + \ln n,$$

which, by reorganizing terms, implies that

$$\mathbb{E}[m] \leq (1 + \eta) \cdot m^* + \frac{\ln n}{\eta},$$

as desired. □

This brings us to a bound which is almost as good as the best expert, modulo a multiplicative loss which can be made arbitrarily small at the cost of paying an additive factor.

Now that we have practiced several iterations of MWU-type algorithms and how to analyze them, we are ready to use this technique for designing algorithms for some classical problems in LPs. This will be the topic of the next chapter.