

Lecture 9

Lecturer: Aleksander Mądry

Scribe: Aleksander Mądry

1 Overview

Today, we take a detour and explore a topic that seems unrelated to the theme of the class: decision making under total uncertainty. In particular, we will consider a simple stock market model that captures the task of aggregation of conflicting advice and design optimal algorithms for that setting. Then, we will introduce the general learning from expert advice framework as well as state and analyze the fundamental algorithm for this framework: multiplicative weights update method. This algorithm turns out to have wide-spread applications throughout computer science – most notably, in machine learning, optimization, and game theory, and we will see some of them in this class.

2 Decision Making or How to Get Rich (with Good Advice)

Let us start our discussion with a motivating scenario: a very simplified version of stock market prediction. In this version, we view the stock market as a single evolving index. (One can think of it as, e.g., the value of the Dow Jones or NASDAQ.) In each day (round) t , it can either go or down up by one. Our task is to predict at the beginning of each day which one of these two possibilities will occur. We are interested in minimizing our total number of prediction mistakes over a fixed (but large) number T of days this games repeats.

It is not hard to see that if we are not equipped with any knowledge on the behavior of the stock market, we cannot hope for any gain over random guessing. Such random guessing would make us mis-predict every other round, in expectation. In fact, in the (not that unrealistic) case when we do not have access to true randomness and the market is completely adversarial to us, one might end up mis-predicting in every round.¹

Therefore, to make this model interesting, we introduce a set of n “experts” (or sources of information) such that at the beginning of each round t , each one of them provides us with his/her “prediction” of the behavior of the stock market in that round. (For instance, one can think of these experts as financial advisors from CNN, Wall Street Journal, etc., or even our neighbor or barber.)

Now, the key feature (and difficulty) of this scenario is that these “experts” might be experts in the name only. That is, most – if not all – of these experts might be unreliable and the advice they are feeding us could be random, arbitrarily correlated or straight misleading.

So, our goal here is to develop an algorithm that can still utilize this “noisy” information. Specifically, we want to make fairly good predictions as long as at least one (initially unknown to us) expert is consistently providing good advice. The way we make this desire precise is by focusing on minimizing our *regret* $\text{Regret}(T)$ after T rounds, which is defined here as

$$\text{Regret}(T) := M(T) - \min_i M_i(T), \quad (1)$$

where $M(T)$ (resp. $M_i(T)$) is the number of mis-predictions our algorithm (resp. expert i) made in all T rounds. In other words, we want to minimize the difference between the number of prediction mistakes we made and the number of prediction mistakes made by the expert that was the best one *in hindsight*. (Observe that it does not really matter here what exactly are the two possible predictions in each round. All that matters is whether we chose the “correct” one or not.)

¹Of course, one could argue that the stock market behavior is far from completely random or adversarial and, after all, many market analysts use sophisticated stochastic models to describe it. Still, even if that’s true, we might not have access to these models or they might be too computationally expensive for us to compute. Also, taking this radical, adversarial point of view will make the results that we develop much more broadly applicable.

3 The Weighted Majority Algorithm

Once we described our setting, let us think about what kind of algorithms could we apply to it. If at some round t , we presented with conflicting advice, which decision should we make?

One natural approach is to simply go with the majority advice. However, this is not the best idea. After all, what could happen is that there is one expert who always gives us correct advice and everyone else mis-predicts in each step. Going with majority each time would make us mis-predict in every step making our regret $\text{Regret}(T)$ be T , i.e., maximal possible.

Fortunately, it is not hard to fix the above approach. Roughly speaking, the problem with it is that we always treat each expert in the same way, irregardless of their past performance. In particular, the influence on the decision of an expert who has not made any mistake up to now is exactly the same the influence of an expert that mis-predicted in every hitherto step. Clearly, this is a flaw and we should remedy it somehow.

The most natural solution is to maintain a weight w_i^t for each expert i that change with time and measure how “trustworthy” the expert i is given his/her performance so far. Given these weights, our prediction corresponds then to taking a *weighted* majority of all the experts’ advice. The exact algorithm is presented as Algorithm 1.

Algorithm 1 Weighted majority algorithm

Set $w_i^0 \leftarrow 1$, for each expert i

for $t = 1 \dots T$ **do**

Follow the advice of the weighted majority of experts (wrt weights w_i^{t-1}), breaking ties arbitrarily

Once the actual outcome is revealed, set

$$w_i^t \leftarrow \begin{cases} (1 - \eta)w_i^{t-1} & \text{if the expert } i \text{ made a mistake this round} \\ w_i^{t-1} & \text{otherwise,} \end{cases}$$

where $0 < \eta \leq \frac{1}{2}$ is a parameter of the algorithm

end for

The parameter η above is often referred to as *learning rate*. The larger it is the stronger the penalty for each mistake.

Let us now analyze this algorithm performance.

Lemma 1 For any expert i and any $0 < \eta \leq \frac{1}{2}$, the number of mistakes $M(T)$ of the weighted majority algorithm after T rounds is at most

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

where $M_i(T)$ is the number of mistakes the expert i made after T rounds.

Proof Our argument will be potential based, with our potential function being

$$W^t = \sum_i w_i^t,$$

i.e., the sum of all the weights in round t .

Clearly, $W^0 = n$ and W^t can only decrease with time, i.e.,

$$W^t \leq W^{t-1}, \tag{2}$$

for each $1 \leq t \leq T$.

We want to understand now how much this potential function decreases whenever the algorithm makes a mistake. By definition of the algorithm, if we make a mistake in round t then the total weight

of the experts that were incorrect is at least $W^{t-1}/2$. As their weight is reduced by a factor of $(1 - \eta)$, we must have that

$$W^t \leq \left(1 - \frac{\eta}{2}\right) W^{t-1} \quad (3)$$

in such rounds.

Putting (2) and (3) together, we can conclude that after the last round T it is the case that

$$W^T \leq \left(1 - \frac{\eta}{2}\right)^{M(T)} W^0 = \left(1 - \frac{\eta}{2}\right)^{M(T)} n.$$

On the other hand, by definition of the algorithm, at the end of round T we have that

$$w_i^T = (1 - \eta)^{M_i(T)}.$$

As, obviously,

$$w_i^T \leq W^T,$$

we can use (3) to conclude that

$$(1 - \eta)^{M_i(T)} = w_i^T \leq W^T \leq \left(1 - \frac{\eta}{2}\right)^{M(T)} n.$$

Taking natural logarithm of both side and using the fact that, for any x with $|x| \leq \frac{1}{2}$,

$$-x - x^2 \leq \ln(1 - x) \leq -x, \quad (4)$$

we get that

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

and thus

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

Dividing both sides by $-\frac{\eta}{2}$ and rearranging the terms gives us the desired bound. ■

Observe that the bound in Lemma 1 holds for any choice of the expert i . So, in particular, it holds for the one that is the best in hindsight. Consequently, even though our algorithm does not know who this best expert i^* is upfront it able to achieve a performance that, in the limit of large T , is within a factor of $2(1 + \eta)$ of that expert's performance.

More precisely, if we divide both sides of the bound from Lemma 1 by T , we obtain that

$$\widehat{M} \leq 2(1 + \eta)\widehat{M}_{i^*} + \frac{2 \ln n}{\eta T} \stackrel{T \rightarrow \infty}{\cong} 2(1 + \eta)\widehat{M}_{i^*},$$

where \widehat{M} (resp., \widehat{M}_{i^*}) is the average number of mistakes of our algorithm (resp., expert i^*) over the whole sequence of T rounds.

We thus see that the learning rate η governs the trade-off between the multiplicative approximation bound of $2(1 + \eta)$ – that corresponds to large T regime – and the additive approximation bound of $\frac{2 \ln n}{\eta}$. The larger T is the smaller value of η we should choose, and vice versa. In fact, one can show that we do not even need to know the value of T in advance. There is a way of adaptively changing the learning rate η to guarantee performance that is close to the one offered by the optimal choice of η .

4 Randomized Weighted Majority Algorithm

The performance of the weighted majority algorithm (Algorithm 1) as expressed by Lemma 1 is quite satisfying. However, asymptotically, it only promises being within a factor of 2 of the best expert. Can we improve this guarantee?

It is not hard to show that this factor of 2 slack is unavoidable, at least as long as we insist on our algorithm to be deterministic. However, once our algorithm is allowed to use private randomness, i.e., randomness that is independent of the chosen sequence of “correct” decisions, one can obtain an improved bound.

The improved algorithm – called *randomized weighted majority algorithm* – is just a simple modification of the weighted majority algorithm we have seen before. Namely, we make it take a randomized weighted majority vote, instead of a deterministic one. That is, in each round t we sample one of the experts i proportionally to their current weight, i.e., with probability p_i^t defined as

$$p_i^t := \frac{w_i^{t-1}}{W^{t-1}}, \quad (5)$$

where $W^t := \sum_i w_i^t$ and follow his/her advice. The resulting algorithm is presented as Algorithm 2.

Algorithm 2 Randomized weighted majority algorithm

Set $w_i^0 \leftarrow 1$, for each expert i

for $t = 1 \dots T$ **do**

 Choose an expert i^t with probability p_i^t (defined in (5)) and follow his/her advice

 Once the actual outcome is revealed, set

$$w_i^t \leftarrow \begin{cases} (1 - \eta)w_i^{t-1} & \text{if the expert } i \text{ made a mistake this round} \\ w_i^{t-1} & \text{otherwise,} \end{cases}$$

 where $0 < \eta \leq \frac{1}{2}$ is a parameter of the algorithm

end for

The performance of this algorithm is described by the following lemma.

Lemma 2 For any expert i and any $0 < \eta \leq \frac{1}{2}$, the expected number of mistakes $M(T)$ of the randomized weighted majority algorithm after T rounds is at most

$$E[M(T)] \leq (1 + \eta)M_i(T) + \frac{\ln n}{\eta},$$

where $M_i(T)$ is the number of mistakes the expert i made after T rounds.

Proof The proof is quite similar to the proof of Lemma 1. Again, we will use the potential function $W^t := \sum_i w_i^t$. As before, $W^0 = n$ and

$$w_i^T = (1 - \eta)^{M_i(T)}. \quad (6)$$

Now, let F^t be the *weighted* fraction of experts that made a mistake in round t . Observe that

$$E[M(T)] = \sum_{t=1}^T F_t. \quad (7)$$

We also have that

$$W^T \leq (1 - \eta F_1)(1 - \eta F_2) \dots (1 - \eta F_T)W^0 = \prod_{t=1}^T (1 - \eta F_t)n.$$

Putting this and (6) together, as well as noting again that $w_i^T \leq W^T$, we obtain

$$(1 - \eta)^{M_i(T)} = w_i^T \leq W^T \leq \prod_{t=1}^T (1 - \eta F_t)n$$

Taking a natural logarithm of both sides gives us that

$$M_i(T) \ln(1 - \eta) \leq \ln n + \sum_{t=1}^T \ln(1 - \eta F_t).$$

Using the same Taylor approximation (4) as before, we get that

$$-\eta(1 + \eta)M_i(T) \leq \ln n - \eta \sum_{t=1}^T F_t = \ln n - \eta E[M(T)],$$

where we also used (7).

Dividing both sides by $-\eta$ and rearranging the terms gives us the desired bound once again. ■

Comparing the bound obtained in this lemma with the bound for weighted majority algorithm presented in Lemma 1, we see that we managed to get rid of the extra factor of 2 entirely, at the cost of using randomness and obtaining the desired performance only in expectation.

By our discussion after Lemma 1, we see that in the regime of large T the randomized weighted majority algorithm's performance essentially matches the performances of the expert i^* that is the best one in hindsight. Also, observe that by setting $\eta = \sqrt{\frac{\ln n}{T}}$, Lemma 2 implies that the expected regret (see (1)) of the randomized weighted majority algorithm can be bounded as

$$\begin{aligned} E[\text{Regret}(T)] &= E[M(T)] - M_{i^*}(T) \leq (1 + \eta)M_{i^*}(T) + \frac{\ln n}{\eta} - M_{i^*}(T) \\ &= \eta M_{i^*}(T) + \frac{\ln n}{\eta} \leq \eta T + \frac{\ln n}{\eta} = 2\sqrt{T \ln n}, \end{aligned}$$

where we used a trivial bound that $M_{i^*}(T) \leq T$.

Finally, it is worth noting that the performance bound of the randomized weighted majority algorithm described in Lemma 2 is essentially the best possible one for any (randomized) algorithm.

5 The Learning From Expert Advice Framework

It turns out that the above ideas give rise to a much more general and versatile framework, which that encompasses many more scenarios than just our above stock market toy model. This framework is called *learning from expert advice*.

Here, we again have n “experts” – although now, one might view them as choices/options – and we are playing a T -round repeated game described as Algorithm 3.

Algorithm 3 Learning from expert advice framework

for each round $t = 1 \dots T$ **do**

 Choose a *convex combination* $p^t := (p_1^t, \dots, p_n^t) \in \Delta_n$ of experts

 Once we have made our choice of p^t , a “loss” $l_i^t \in [-1, 1]$ is revealed for each expert i

 Our resulting loss in round t is $l^t := \sum_i p_i^t l_i^t$

end for

Above, $\Delta_n \subset \mathbb{R}^n$ denotes the n -dimensional simplex, i.e., a set of n -dimensional points whose coordinates are non-negative and sum up to 1. (Thus, any point in the simplex corresponds to a probability distribution over n objects.)

As before, our goal is to devise a strategy for choosing the convex combinations p^t in each step t , so that the total loss we incur is not much worse than the total loss of the best-in-hindsight expert. More precisely, we again want to minimize the regret

$$\text{Regret}(T) := L(T) - \min_i L_i(T), \tag{8}$$

where $L(T)$ (resp., $L_i(T)$) is the total loss $\sum_{t=1}^T l^t$ of our algorithm (resp., the total loss $\sum_{t=1}^T l_i^t$ of the expert i).

Observe that in this framework we allow l_i^t to be negative – we interpret such negative loss as gain. Also, as we already mentioned, the convex combinations p^t can be directly interpreted as a probability distributions. So, by taking a random choice according to them in each round, one can get the same performance bound in expectation. This is important in some applications where taking a convex combination of the available experts is not feasible.

Finally, note that this framework indeed generalizes our toy stock market model. One just needs to take l_i^t to be 1 if the expert i makes a wrong prediction in round t and l_i^t to be 0 otherwise. Clearly, our total (expected) loss will be equal to the expected number of mistakes we make.

6 The Multiplicative Weights Update Algorithm

Once the general learning from expert advice framework is setup, we are able to lift the randomized weighted majority algorithm to this setting. The resulting algorithm is called the *multiplicative weights update method* and is presented as Algorithm 4

Algorithm 4 Multiplicative weights update method

Set $w_i^0 \leftarrow 1$, for each expert i

for $t = 1 \dots T$ **do**

Choose $p_i^t := \frac{w_i^{t-1}}{W^{t-1}}$, for each expert i , i.e., each expert is taken proportionally to his/her weight.
Once the losses l_i^t for all the experts i are revealed, set

$$w_i^t := (1 - \eta l_i^t) w_i^{t-1},$$

where $0 < \eta \leq \frac{1}{2}$ is a parameter of the algorithm

end for

Observe that, since we have always that $l_i^t \in [-1, 1]$, each multiplicative update $(1 - \eta l_i^t)$ to the weights w_i^{t-1} is a factor between $(1 - \eta)$ and $(1 + \eta)$.

Now, the performance of the MWU algorithm is provided by the following theorem.

Theorem 3 For any $0 < \eta \leq 1/2$ and any expert i , the total loss $L(T)$ of the multiplicative weights update method (Algorithm 4) after T rounds is

$$L(T) \leq L_i(T) + \eta \sum_{t=1}^T |l_i^t| + \frac{\ln n}{\eta},$$

where $L_i(T) = \sum_{t=1}^T l_i^t$ is the total loss of expert i after T rounds.

Observe that, again, the performance above means that our algorithm matches the performance of the best expert when $T \rightarrow \infty$. In particular, if all the losses l_i^t are non-negative then the above bound becomes

$$L(T) \leq L_i(T) + \eta \sum_{t=1}^T |l_i^t| + \frac{\ln n}{\eta} = L_i(T) + \eta \sum_{t=1}^T l_i^t + \frac{\ln n}{\eta} = (1 + \eta)L_i(T) + \frac{\ln n}{\eta},$$

which is an expression analogous to what we obtained in Lemma 2 for the randomized weighted majority algorithm.

Also, setting $\eta = \sqrt{\frac{\ln n}{T}}$ in the bound from Theorem 3 gives us a bound on regret

$$\text{Regret}(T) = L(T) - \min_i L_i(T) \leq \eta \sum_{t=1}^T |l_i^t| + \frac{\ln n}{\eta} \leq \eta T + \frac{\ln n}{\eta} = 2\sqrt{T \ln n}$$

that is analogous to the bound we obtained for randomized weighted majority algorithm.

Finally, similarly to the case of the randomized weighted majority algorithm, the performance of the multiplicative weights update method can be shown to be essentially best possible.

Let us now prove the theorem.

Proof Once again, the proof relies on understanding the evolution of the quantities W^t and w_i^t for our fixed expert i .

To understand the former, notice that $W^0 = n$. Also, in each round t , we have

$$W^t = \sum_i w_i^t = \sum_i (1 - \eta l_i^t) w_i^{t-1} = W^{t-1} - \eta W^{t-1} \sum_i p_i^t l_i^t = (1 - \eta l^t) W^{t-1},$$

where l^t is the loss of the algorithm in round t . So, we have that

$$W^T = \prod_{t=1}^T (1 - \eta l^t) n. \tag{9}$$

The evolution of w_i^t is also very simple, we have $w_i^0 = 1$ and then

$$w_i^T = \prod_{t=1}^T (1 - \eta l_i^t).$$

Now, we can just lower bound W^T by w_i^T to obtain that

$$\prod_{t=1}^T (1 - \eta l_i^t) = w_i^T \leq W^T \leq \prod_{t=1}^T (1 - \eta l^t) n.$$

Taking a natural logarithm of both sides and using our Taylor approximation (4) we get that

$$-\eta \sum_{t=1}^T (l_i^t + \eta (l_i^t)^2) \leq -\eta \sum_{t=1}^T l^t + \ln n = -\eta L(T) + \ln n.$$

Rearranging the terms, dividing both sides by $-\eta$ and using the fact that $(l_i^t)^2 \leq |l_i^t|$, we obtain

$$L(T) \leq L_i(T) + \eta \sum_t |l_i^t| + \frac{\ln n}{\eta},$$

as desired. ■