Lecture 4

1 The multiplicative weights update method

The multiplicative weights method is a very useful framework applicable to a wide variety of learning and optimization problems. The method was first discovered in the context of online learning, and has been rediscovered several times since then.

Algorithms in the multiplicative weights framework maintain a probability distribution/ weights over a certain set that is updated iteratively by a multiplicative rule. The analysis of these algorithms relies on quantifying the change in an exponential potential function.

1.1 An Infallible Expert

The simplest example illustrating multiplicative weights is the following: There are n experts $E_1, \ldots, E_n$ who predict the stock market every day. The predictions of the experts are binary valued (up/down). It is known that at least one of the experts never makes a mistake.

An online learning algorithm sees the predictions of the experts every day and makes a prediction of its own. The goal of the online algorithm is to minimize the total number of mistakes made. The following algorithm makes at most $\log n$ mistakes:

1. Initialize the set of experts who have not yet made a mistake to $E = \{E_1, E_2, \ldots, E_n\}$.
2. Predict according to the majority of experts in the set E.
3. Update E by removing all the experts who predicted incorrectly.

Invariant: If the algorithm makes a mistake, at least half the experts in E are wrong. The size of E gets reduced by at least $1/2$ for every incorrect prediction, so the total number of mistakes made is at most $\log n$. The bound on the number of mistakes is independent of the number of days, the algorithm does not make more than $\log n$ mistakes.

The presence of an infallible expert ensures that E is non empty, so the algorithm can always make a prediction. This example is based on the unrealistic assumption that some expert is always correct, but the following features should be noted: (i) The algorithm maintains weights on a set of experts which are updated iteratively, the update rule being multiplication by 0 for all wrong experts. (ii) If the algorithm makes a mistake, something drastic happens, the size of the set E is reduced by $1/2$.

1.2 Imperfect Experts and the Weighted Majority Algorithm

The previous algorithm can not handle of imperfect experts as E would eventually be empty and no prediction can be made. Experts making wrong predictions should not be dropped, but their weights can be reduced by a constant factor, say $1/2$. The modified algorithm is:

1. Initialize $w_i = 1$, where w_i is the weight of the i-th expert.
2. Predict according to the weighted majority of experts.
3. Update weights by setting $w_i \leftarrow w_i/2$ for all experts who predicted incorrectly.

Invariant: If the algorithm makes a mistake, the weight of the wrong experts is at least half the total weight. The weight of the wrong experts gets reduced by least $1/2$, so the total weight is reduced by a factor of at least $3/4$ for every mistake. After making M mistakes the total weight is at most $n.\left(\frac{3}{4}\right)^M$.

The potential function for the analysis is W, the total weight of the experts. If the best expert makes m mistakes, the total weight W must be at least $1/2^m$. Combining the upper and lower bounds,

$$\frac{1}{2^m} \leq W \leq n\left(\frac{3}{4}\right)^M$$

Taking logarithms we have a worst case bound on the total number of mistakes M made by the algorithm,

$$-m \leq \log n + M\log(3/4) \Rightarrow M \leq \frac{m+\log n}{\log(4/3)} \leq 2.4(m+\log n) \tag{1}$$

The weighted majority algorithm does not make too many more mistakes compared to the best expert. The mistake bound can be improved by using a multiplicative factor of $(1-\epsilon)$ in the experts algorithm. The following standard approximations will be used for the analysis,

PROPOSITION 1
For $\epsilon \in [0, 1/2]$ we have the approximation,

$$-\epsilon - \epsilon^2 \leq \ln(1-\epsilon) < -\epsilon$$

PROOF: The Taylor series expansion for $\ln(1-\epsilon)$ is given by,

$$\ln(1-\epsilon) = \sum_{i\in\mathbb{N}} -\frac{\epsilon^i}{i}$$

From the expansion we have $\ln(1-\epsilon) < -\epsilon$ as the discarded terms are negative. The other half of the inequality is equivalent to the inequality $1-\epsilon \geq e^{-\epsilon-\epsilon^2}$ for $\epsilon \in [0, 1/2]$. By the convexity of the function e^{-x-x^2}, the inequality is true for all ϵ less than a threshold t. Substituting $\epsilon = 1/2$ we have $1/2 \geq e^{-3/4} = 0.47$ showing that the threshold t is more than $1/2$. □

PROPOSITION 2
For $\epsilon \in [0, 1]$ we have,

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

PROOF: Note that $(1-\epsilon)^x$ is a convex function so there exists t such that $(1-\epsilon)^x \leq (1-\epsilon x)$ if and only if $x \in [0, t]$. Equality holds for $x = 1$ so the threshold t is 1. The other inequality is proved similarly. □

1.3 Analysis with multiplicative factor $1-\epsilon$

Claim 3
The number of mistakes M made by the experts algorithm with multiplicative factor of $(1-\epsilon)$ is bounded by,

$$M \leq 2(1+\epsilon)m + \frac{2\ln n}{\epsilon} \tag{2}$$

Proof: In this case, we have that the best expert has weight at least $(1-\epsilon)^m$. Moreover, in each step where a mistake is made, at least half the weight is reduced by a factor of $(1-\epsilon)$, which implies that after M mistakes the weight is at most $(1-\epsilon/2)^M n$. Thus, we have the following bound on the total weight,

$$(1-\epsilon)^m \leq W \leq (1-\epsilon/2)^M n$$

Taking logs we have,

$$m\ln(1-\epsilon) \leq \ln n + M\ln(1-\epsilon/2)$$

The approximation from proposition 1 is used on both sides of the inequality to replace the $\ln(1-\epsilon)$s by expressions depending on ϵ,

$$-m(\epsilon+\epsilon^2) \leq \ln n - M\epsilon/2$$

Rearranging and dividing by $\epsilon/2$ we have the mistake bound,

$$M \leq 2m(1+\epsilon) + \frac{2\ln n}{\epsilon}$$

□

The constant in the bound (2) is better than that in (1) but the following example shows that it is not possible to achieve a constant better than 2 with the weighted majority strategy. Suppose there are two experts A and B where A is right on odd numbered days while B is right on even numbered days. For all ϵ, the weighted majority algorithm makes incorrect predictions after round 1 as the incorrect expert gets assigned more than $1/2$ the weight. The algorithm always predicts incorrectly, while the best experts is wrong half the time showing that the factor of 2 is tight.

A probabilistic strategy that chooses experts with probabilities proportional to their weights performs better than the deterministic weighted majority rule. The expected number of mistakes made by the probabilistic strategy is,

$$M \leq (1+\epsilon)m + \frac{\ln n}{\epsilon} \tag{3}$$

1.4 Probabilistic experts algorithm

We generalize the setting by allowing the losses suffered by the experts to be real numbers in $[0,1]$ instead of binary values. The loss suffered by the i-th expert in round t is denoted by $\ell_i^{(t)} \in [0,1]$. The probabilistic algorithm is the following:

1. Initialize $w_i = 1$, where w_i is the weight of the i-th expert.
2. Predict according to an expert chosen with probability proportional to w_i, the probability of choosing the i-th expert is $\frac{w(i)}{W}$ where W is the total weight.
3. Update weights by setting $w_i \leftarrow w_i(1-\epsilon)^{\ell_i^{(t)}}$ for all experts.

CLAIM 4
If L is the expected loss of the probabilistic experts algorithm and L^* is the loss of the best expert then,

$$L \leq \frac{\ln n}{\epsilon} + (1+\epsilon)L^* \tag{4}$$

PROOF: The potential function $W(t) := \sum_i w_i$ is the sum of the weight of the experts for round t. The expected loss suffered by the algorithm during round t is given by,

$$L_t = \frac{\sum_i w_i \ell_i^{(t)}}{W(t)}$$

$W(t+1)$ can be calculated directly as the weight w_i gets updated to $w_i(1-\epsilon)^{\ell_i^{(t)}}$ which is less than $w_i(1-\epsilon\ell_i^{(t)})$ from proposition 2 as all the losses are in $[0,1]$. [If the loss belongs to $[0,\rho]$, the update rule is $w_i \to (1-\epsilon)^{l_i/\rho}$, the analysis yields $L \leq \frac{\rho \ln n}{\epsilon} + (1+\epsilon)L^*$].

$$\begin{aligned} W(t+1) \leq \sum_i (1-\epsilon\ell_i^{(t)})w_i &= \sum_i w_i - \epsilon \sum_i w_i \ell_i^{(t)} \\ &= \sum_i w_i \left(1 - \epsilon \frac{\sum_i w_i \ell_i^{(t)}}{\sum_i w_i}\right) \\ &= W(t)(1-\epsilon L_t) \end{aligned}$$

The initial value of the potential function is $W(0) = n$, and so the final value can be bounded by,

$$W(T) \leq n \prod_{t\in[T]} (1-\epsilon L_t) \tag{5}$$

If the best expert incurs a total loss of L^*, then $W(T) \geq (1-\epsilon)^{L^*}$ as the total weight is at least the weight of the best expert. Combining the upper and lower bounds and taking logarithms we have,

$$\begin{aligned} (1-\epsilon)^{L^*} &\leq n \prod_{t\in[T]} (1-\epsilon L_t) \\ \Rightarrow L^* \ln(1-\epsilon) &\leq \ln n + \sum_{t\in[T]} \ln(1-\epsilon L_t) \end{aligned} \tag{6}$$

Using the approximation from proposition 1 to replace the $\ln(1-\epsilon)$s by expressions depending on ϵ,

$$-L^*(\epsilon + \epsilon^2) \leq \ln n - \epsilon \sum_{t \in T} L_t \tag{7}$$

Rearranging and using the fact that the expected loss $L = \sum_{t \in T} L_t$ we have,

$$L \leq \frac{\ln n}{\epsilon} + L^*(1 + \epsilon) \tag{8}$$

□

Exercise: If we run the multiplicative weights algorithm with gains $g_i \in [0, 1]$ updating the weight of an expert i using the rule $w_i.(1 + \epsilon)^{g_i}$, a similar analysis yields,

$$G \geq (1 - \epsilon)G^* - \frac{\ln n}{\epsilon} \tag{9}$$

2 Wrap-Up

The multiplicative weights method is very simple way to achieve provably good bounds in the sense of doing as well as the best expert in retrospect. In future lectures we will see the the multiplicative weights method applied to problems like finding ϵ optimal strategies for zero sum games and boosting.