Lectures 5-6

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 perhaps first discovered in the fifties in the context of game theory, and was rediscovered in the context of online learning, and has been rediscovered several times since then.

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). We consider the case where 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 $O(\log n)$ 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.

Analysis: 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 mistake, so the number of mistakes made is at most $O(\log n)$. The bound on the number of mistakes is independent of the number of days.

The presence of an infallible expert ensures that E is non empty, so the algorithm can always make a prediction. This example is rather unrealistic, but the following features should be noted: (i) The algorithm may be regarded as maintaining weights on a set of experts, the weights are $0/1$, the update rule is to multiply by 0 the weights of all experts who make an incorrect prediction. (ii) If the algorithm makes a mistake, something drastic happens, the total weight gets reduced by $1/2$.

1.2 Imperfect Experts and the Weighted Majority Algorithm

The previous algorithm can not handle imperfect experts as E will eventually be empty and no prediction can be made. To deal with this situation, experts making incorrect 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.

Analysis: We use the total weight of the experts $W = \sum_i w_i$ as a potential function to keep track of the performance of the algorithm. The initial value of W is n, and whenever the algorithm makes a mistake, the weight of the incorrect experts is at least half the total weight. The weight of the incorrect experts gets reduced by least 1/2, so the total weight is reduced by a factor of at least 3/4 for every mistake. If the algorithm makes M mistakes the final value of W is at most $n\left(\frac{3}{4}\right)^M$.

The total weight is at least the weight of the best expert, so if the best expert makes m mistakes, 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 $(1-\epsilon)$ instead of 1/2 in the experts algorithm.

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

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

The proof is similar to the $\epsilon = 1/2$ case discussed above and is left as an exercise for the reader.

The following example shows that it is not possible to achieve a constant better than 2 in bound (2) using 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 weighted majority rule for the above example. We will show that the expected number of mistakes made by the probabilistic strategy is,

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

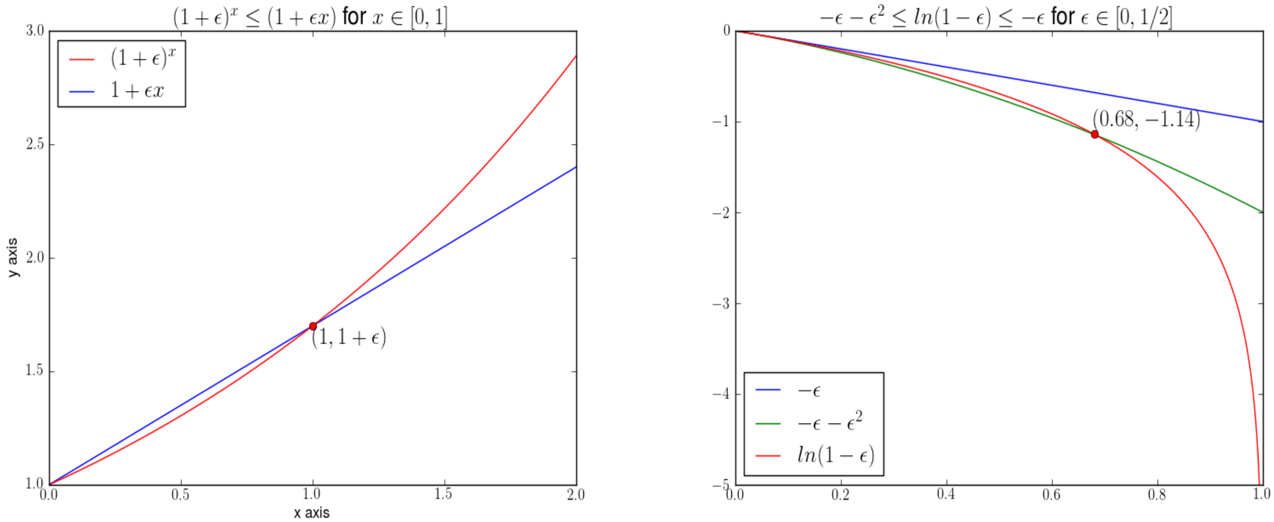

Figure 1: Proof in pictures of approximations used in the experts algorithm.

1.3 Approximations

The analysis of the probabilistic experts algorithm will require the following standard approximations, the proofs follow by the convexity of the exponential function and are illustrated in Figure 1.

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

$$\begin{aligned} -\epsilon - \epsilon^2 &\leq \ln(1-\epsilon) < -\epsilon \\ \epsilon - \epsilon^2 &\leq \ln(1+\epsilon) < \epsilon \end{aligned}$$

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 such that $0 \leq \epsilon \leq t$, for some 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 3
For $\epsilon \in [0, 1]$ we have,

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

1.4 Probabilistic experts algorithm

We generalize the setting by allowing 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 experts 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 analysis of the experts algorithm relies on two observations quantifying the change in the total weight $W(t) := \sum_i w_i$. First we note that the initial value $W(0) = n$ and the final value $W(T)$ is at least the weight of the best expert. If the best expert suffers loss L^*,

$$n(1-\epsilon)^{L^*} \leq W(T) \tag{5}$$

The second observation is that the decrease in weight over a single round is bounded by the expected loss of the experts algorithm,

$$W(t+1) \leq W(t)(1-\epsilon L_t) \tag{6}$$

PROOF: The expected loss suffered by the algorithm during round t is $L_t = \frac{\sum_i w_i \ell_i^{(t)}}{W(t)}$.

The weight w_i gets updated to $w_i(1-\epsilon)^{\ell_i^{(t)}}$ which is less than $w_i(1-\epsilon\ell_i^{(t)})$ by proposition 2 as all the losses belong to $[0,1]$.

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

□

Combining the two observations we have,

$$(1-\epsilon)^{L^*} \leq W(T) \leq n \prod_{t\in[T]} (1-\epsilon L_t) \tag{7}$$

Taking logarithms,

$$L^* \ln(1-\epsilon) \quad \leq \quad \ln n + \sum_{t\in[T]} \ln(1-\epsilon L_t)$$

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

$$-L^*(\epsilon+\epsilon^2) \leq \ln n - \epsilon \sum_{t\in T} L_t \tag{8}$$

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

$$L \leq \frac{\ln n}{\epsilon} + L^*(1+\epsilon) \tag{9}$$

□

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,

$$G \geq (1-\epsilon)G^* - \frac{\ln n}{\epsilon} \tag{10}$$

If the loss belongs to $[0,\rho]$ and the update rule is $w_i \leftarrow (1-\epsilon)^{l_i/\rho} w_i$, the analysis yields:

$$L \leq \frac{\rho \ln n}{\epsilon} + (1+\epsilon)L^* \tag{11}$$

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 apply the the multiplicative weights method to problems like finding ϵ optimal strategies for zero sum games and boosting.