18.409 An Algorithmist's Toolkit December 8, 2009

Lecture 24

Lecturer: Jonathan Kelner *Scribe: Dimiter Ostrev*

Multiplicative Weights

In this lecture we will introduce Multiplicative Weights, a simple technique with many applications. We start with an example.

Example Suppose Mr. X wants to bet on football games but does not know much about football himself. Before each game, X can check the predictions of n experts. Is there an algorithm that allows Mr. X to perform well in the long run?

Two potential ideas are:

(1) For each game, bet according to what the majority of experts predict

(2) Wait a few games to see which of the experts get it right most of the time and then follow their advice

These strategies work well in some cases but not in others: (1) fails when only a few experts make good predictions, and (2) fails when there is an expert that performs well for the first few games and then never makes a correct prediction again. Instead, we will consider a combination of the two approaches: for each game, we will consider the opinion of all experts, but each expert's opinion will be weighted according to his past performance. More precisely, let w_i^t denote the weight of expert i after t games, and consider the following algorithm:

1. Set $w_i^0 = 1$ for $i = 1, ..., n$
2. Make a prediction for game t based on a weighted majority of experts where expert i gets weight $w_i^{t-1} / \sum_j w_j^{t-1}$
3. After game t update the weights as follows: if expert i's prediction for game t was wrong then set $w_i^t = (1-\epsilon)w_i^{t-1}$; otherwise set $w_i^t = w_i^{t-1}$

For this algorithm, we have the following:

Theorem Let m_i^t denote the number of mistakes that expert i makes in the first t games and m^t denote the number of mistakes that Mr. X makes in the first t games. Then for all i and t,

$$m^t \leq \frac{2log(n)}{\epsilon} + 2(1+\epsilon)m_i^t$$

and in particular, this holds for the i that minimizes m_i^t.

Proof Define $\Phi^k = \sum_i w_i^k$. If Mr. X makes a mistake at game k, then a weighted majority of the experts must have made a wrong prediction for game k. The weights of all these experts drop by a factor of $(1-\epsilon)$ and so we have $\Phi^k \leq (1-\epsilon/2)\Phi^{k-1}$. Then over the first t games we have

$$\Phi^t \leq (1-\frac{\epsilon}{2})^{m^t}\Phi^0 = n(1-\frac{\epsilon}{2})^{m^t}$$

On the other hand we have $w_i^t = (1-\epsilon)^{m_i^t}$ and so

$$\Phi^t \geq w_i^t = (1-\epsilon)^{m_i^t}$$

Therefore,

$$n(1-\frac{\epsilon}{2})^{m^t} \geq (1-\epsilon)^{m_i^t}$$

Rearranging this inequality gives

$$m^t \leq \frac{log(n)}{-log(1-\epsilon/2)} + m_i^t \frac{log(1-\epsilon)}{log(1-\epsilon/2)}$$

This bound is slightly stronger than the one in the statement of the theorem. Using the inequalities $\epsilon/2 \leq -log(1-\epsilon/2)$ and $\epsilon + \epsilon^2 \geq -log(1-\epsilon)$ converts it to the required form and completes the proof.

Next, we will modify our algorithm to get rid of the factor of 2 on the right hand side of the bound above. Consider the following:

1. Set $w_i^0 = 1$ for $i = 1, ..., n$
2. To make a prediction for game t, do the following: for $i = 1, ...n$, follow expert i's prediction with probability $p_i^t = w_i^{t-1} / \sum_j w_j^{t-1}$
3. After game t update the weights as follows: if expert i's prediction for game t was wrong then set $w_i^t = (1-\epsilon)w_i^{t-1}$ else set $w_i^t = w_i^{t-1}$

For this algorithm, we have the following:

Theorem Let m_i^t denote the number of mistakes that expert i makes in the first t games and let m^t denote the random variable equal to the number of mistakes that Mr. X makes in the first t games. Then for $\epsilon < 1/2$ and for all i and t,

$$E(m^t) \leq \frac{log(n)}{\epsilon} + (1+\epsilon)m_i^t$$

and in particular, this holds for the i that minimizes m_i^t.

The proof of this Theorem is similar to before and we will omit it. Instead, we will introduce our most general version of the multiplicative weights algorithm. In the example above, we had only two possibilities for the relation between event outcomes and expert predictions: the outcome of game t either matched expert i's prediction or it did not. Our measure of performance for individual experts and for the algorithm as a whole was simply counting wrong predictions. We want to generalize the algorithm to allow for an arbitrary set P of possible outcomes to events. In this setting, we will measure the performance of the algorithm as follows: we will say that at each step, following expert i's prediction when the true outcome is j incurs a penalty of $M(i, j)$. More precisely, we have the following:

0. The input of the algorithm consists of: a set P of possible outcomes to events. For $i = 1, ...n$ and for $j \in P$ a number $M(i, j)$ from the interval $[-l, \rho]$. We will refer to ρ as the width; we will also have the restriction $l < \rho$.
1. Set $w_i^0 = 1$ for $i = 1, ..., n$
2. To make a prediction for event t, do the following: for $i = 1, ...n$, follow expert i's prediction with probability $p_i^t = w_i^{t-1} / \sum_j w_j^{t-1}$

3. Let j^t denote the outcome of event t. Update the weights as follows:

$$w_i^t = \{ \begin{array}{ll} w_i^{t-1}(1-\epsilon)^{M(i,j^t)/\rho} & if\, M(i,j^t) \geq 0 \\ w_i^{t-1}(1+\epsilon)^{-M(i,j^t)/\rho} & if\, M(i,j^t) < 0 \end{array}$$

A similar analysis to before gives:

Theorem Let D^t denote the probability distribution $\{p_1^t, \ldots, p_n^t\}$ with which we pick experts to make a prediction for event t. Let $M(D^t, j^t)$ denote the expected value of our penalty when following the distribution D^t for event t and when the actual outcome is j^t. Then for $\epsilon \leq 1/2$ and for all T and i,

$$\sum_{t=1}^{T} M(D^t, j^t) \leq \frac{\rho log(n)}{\epsilon} + (1+\epsilon) \sum_{t:M(i,j^t)\geq 0} M(i,j^t) + (1-\epsilon) \sum_{t:M(i,j^t)<0} M(i,j^t)$$

Corollary For any δ, for $\epsilon \leq min(1/2, \delta/4\rho)$, for $T = 16\rho^2 log(n)/\delta^2$ rounds and for all i, the average penalty we get per round obeys:

$$\frac{\sum_{t=1}^{T} M(D^t, j^t)}{T} \leq \delta + \frac{\sum_{t=1}^{T} M(i, j^t)}{T}$$

and in particular our average penalty per round is at most δ bigger than the average penalty of the best expert.

Applications of Multiplicative Weights

Our first application of the Multiplicative Weights algorithm will be to zero-sum games. In a zero-sum game, we have a row player, R, and a column player, C. If R plays strategy i and C plays strategy j, then R pays C the amount $M(i,j)$. Players can also play mixed strategies, i.e. probability distributions over the sets of pure strategies. We will extend our payoff notation so that $M(D,P)$ denotes the expected amount that R pays C when R plays the mixed strategy D and C plays the mixed strategy R. Recall that von Neumann's Minimax Theorem states that

$$min_D\; max_j\; M(D,j) = max_P\; min_i M(i,P)$$

We will denote the above quantity by λ; it is known as the value of the game.

We are now ready to state the zero-sum game problem: given the sets of strategies for R and C and the payoffs $M(i,j)$, estimate the value of the game λ. Our approach will be to associate elements of the current problem to appropriately chosen elements of the Multiplicative Weights algorithm, then directly apply what we already know about Multiplicative Weights to conclude that we do indeed get a good approximation to λ in a reasonable amount of time. The details of the argument will be presented next lecture.