THE MULTIPLICATIVE WEIGHT UPDATES METHOD

URI ZWICK AND HAIM KAPLAN

1. The basic Binary Setting

On each of T days, n experts predict "up" or "down", and we have have to make a decision according to their predictions.

1.1. **The Weighted Majority Algorithm.** We choose a parameter $0 \leq \eta \leq \frac{1}{2}$ and assign each expert i an initial weight of $w_i^t = 1$. At the end of the day we decide "up" if and only if $\sum_{\text{i predicted up}} w_i^t \geq \sum_{\text{i predicted down}} w_i^t$. Then we update the weights:

$$w_i^{t+1} = \begin{cases} w_i^t & \text{i was correct on day t} \\ (1-\eta)w_i^t & \text{else} \end{cases}$$

We also define the cost of each expert at day T to be the number of mistakes he did until day T. The cost of the algorithm is defined simmilarily.

Theorem 1. *For every expert it holds that:*

$$cost^T(WM_\eta) \leq 2(1+\eta)cost^t(exp) + \frac{2\ln(n)}{\eta}$$

Proof. Let $W^t = \sum_{i=1}^n w_i^t$. If $MW\eta$ makes a mistake in day t, then the weighted majority of the experts predicted wrong. Therefore,

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

□

Since $W^1 = n$ we have (by induction):

$$W^{t+1} \leq n(1-\frac{\eta}{2})^{cost^t(WM_\eta)}$$

The same analysis yields that: $W^{T+1} \geq w_{exp}^{t+1} = (1-\eta)^{cost^t(exp)}$.

$$(1-\eta)^{cost^t(exp)} \leq W^{t+1} \leq \left(1-\frac{\eta}{2}\right)^{cost^t(WM_\eta)}$$

$$cost^t(exp)\ln(1-\eta) \leq cost^t(WM_\eta)\ln\left(1-\frac{\eta}{2}\right) + \ln(n)$$

$$cost^t(WM_\eta) \leq \frac{\ln(1-\eta)}{\ln\left(1-\frac{\eta}{2}\right)}cost^t(exp) + \frac{\ln(n)}{-\ln(1-\frac{\eta}{2})}$$

$$cost^t(WM_\eta) \leq \frac{\eta+\eta^2}{\frac{\eta}{2}} + \frac{\ln(n)}{\frac{\eta}{2}} = 2(1+\eta)cost^t(exp) + \frac{2\ln(n)}{\eta}$$

Notes written by Bar Aschner,

On the last inequallity we used the fact that $x \leq -\ln(1-x) \leq x + x^2$ for $x \in \left(0, \frac{1}{2}\right)$.

2. The General Setting

On each day we choose a distribution over the experts. Then the cost of each expert (which is a real number in $[-1, 1]$) is revealed and we pay the expected cost according to the distribution chosen.

We start by picking a parameter $\eta \in \left[0, \frac{1}{2}\right]$. The weight of each expert i at day t is w_i^t . We also get the distribution over the experts:

$$p^t = \frac{(w_1^t \dots w_n^t)}{W^t}$$

Where $W^t = \sum_i w_i^t$. m_i^t is the cost of expert i at day t. At the end of the day we update the costs by getting $w_i^{t+1} = w_i^t(1 - \eta m_i^t)$. The cost for the algorithm at day t is $p^t \cdot m^t$. We define the cost of each expert and the cost of the algorithm to be the sum of the costs up to day t. That is, $cost^t(MW_\eta) = \sum_{t=1}^{T} p^t \cdot m^t$ and $cost(exp) = \sum_{t=1}^{T} m_{exp}^T$.

Theorem 2. *For every expert it holds that:*

$$cost^t(MW_\eta) \leq cost^t(exp) + \eta \sum_{t=1}^{T} (m_i^t)^2 + \frac{\ln(n)}{\eta}$$

Proof.

$$\ln \frac{W^{t+1}}{W^1} = \sum_{t=1}^{T} \ln \frac{W^{t+1}}{W^t} = \sum_{t=1}^{T} \ln \left(\sum_{i=1}^{n} p_i^t (1 - \eta m_i^t) \right) = \sum_{t=1}^{T} \ln(1 - \eta p^t \cdot m^t) \leq -\eta \sum_{t=1}^{T} p^t \cdot m^t$$

□

On the last inequallity we used the fact that $\ln(1-x) \leq -x$ for $x \leq \frac{1}{2}$.

On the other hand:

$$\ln \frac{W^{t+1}}{W^1} \geq \frac{w_{exp}^{t+1}}{n} = \sum_{t=1}^{T} \ln(1 - \eta m_i^t) - \ln(n)$$

$$\geq \ln(n) - \eta \sum_{t=1}^{T} m_i^t - \eta^2 \sum_{t=1}^{T} (m_i^t)^2$$

We get the result by combining these inequalities.

Corollary 3. *If p is a distribution and $|m^t|$ is the vector obtained by taking the absolute value in each coordinate. Then:*

$$\sum_{t=1}^{T} p^t m^t \leq \sum_{t=1}^{T} p \cdot m^t + \eta \sum_{t=1}^{T} p \cdot |m^t| + \frac{\ln(n)}{\eta}$$

If we have a reward r^t instead of cost, and updated by using the rule $w_i^{t+1} = w_i^t(1 + \eta r_i^t)$ then:

$$\sum_{t=1}^{T} p^t \cdot r^t \geq \sum_{t=1}^{T} r_i^t - \eta \sum_{t=1}^{T} \left(r_i^t\right)^2 - \frac{\ln}{\eta}$$

3. Learning a Linear Classfier

We are given a set of m points $a_1 \dots a_m \in \mathbb{R}^n$. Suppose that there is a **non-negative** vector $x \in \mathbb{R}^n$ such that $a_j \cdot x \geq \epsilon$ for all j. The algorithm finds a **non-negative** vector $u \in \mathbb{R}^n$ such that $1 \cdot u = 1$ and $a_j \cdot u \geq 0$ for all j.

The Algorithm: We define $\rho = \max_j |a_j|_\infty$ .Select $\eta = \frac{\epsilon}{2\rho}$. We will have an "expert" for each coordinate. Run the MW_η algorithm. In each iteration, if p^t is a good classifier, stop. Otherewise, let j be the first index such that $p^t \cdot a_j < 0$. Let $m^t = -\frac{a_j}{\rho}$.

Theorem 4. *This algorithm stops after at most* $\frac{4\rho^2}{\epsilon^2} \ln(n)$ *iterations.*

Proof.

$$\sum_{t=1}^{T} p^t m^t \leq \sum_{t=1}^{T} x \cdot m^t + \eta \sum_{t=1}^{T} x \cdot |m^t| + \frac{\ln(n)}{\eta}$$

□

For every day $t \leq T$ we have a point $a(t)$ such that $a_j(t) \cdot p^t < 0$ and $m^t = -\frac{a(t)}{\rho}$.

$$\sum_{t=1}^{T} p^t \cdot \frac{-a_j(t)}{\rho} \leq \sum_{t=1}^{T} x \cdot \frac{-a(t)}{\rho} + \eta \sum_{t=1}^{T} x \cdot \frac{|a(t)|}{\rho} + \frac{\ln(n)}{\eta}$$

$$0 \leq -\frac{\epsilon T}{\rho} + \eta T + \frac{\ln(n)}{\eta}$$

$$\eta T \leq \frac{\ln(n)}{\eta}$$

$$T \leq \left(\frac{2\rho}{\epsilon}\right)^2 \ln(n)$$

As required.

4. Zero Sum Games

We talk about 2-player (ROW and COLUMN) games with randomized (mixed) startegies. Let A be a matrix. ROW has a disribution p over the rows of A, COLUMN has a distribution q over the columns of A. The expected payoff (row pays column) is $A[p,q] = p^t A q = \sum_{i,j} p_i q_j A[i,j]$.

4.1. Von Neumann's Theorem.

It holds that:

$$\min_p \max_q A[p,q] = \max_q \min_p A[p,q] = \min_p \max_j A[p,j] = \max_q \min_i A[i,q]$$

4.2. Solving Zero-Sum Games Approximately.

We want to approximate the game's value and the optimal strategies. We assume that $A_{ij} \in [0,1]$ for all i,j. Let $v^* = val(A)$ and $\epsilon > 0$. p,q are ϵ-optimal strategies if $\max_j A[p,j] \leq v^* + \epsilon$ and $\min_i A[i.q] \geq v^* - \epsilon$. We have an expert for each of the n rows of A. In each iteration t, the algorithm produces a distribution p^t. The cost vector m^t is the column j^t of A which maximizes $A[p^t, j]$. Note that: $p \cdot m^t = A[p^t, j^t] \geq v^*$.

Theorem 5. *If MW_η is run with $\eta = \frac{\epsilon}{2}$ for $\frac{4\ln(n)}{\epsilon^2}$ iterations, then the best strategy obtained is ϵ optimal for ROW. If A has m columns then the running time is* $O\left(\frac{mn\ln(n)}{\epsilon^2}\right)$.

Proof. First, we bound the running time of the algorithm.

$$\sum_{t=1}^{T} A(p^t, j^t) \leq (1+\eta)\sum_{t=1}^{T} A(p^*, j^t) + \frac{\ln(n)}{\eta}$$

$$v^* \leq \frac{1}{T}\sum_{t=1}^{T} A(p^t, j^t) \leq v^* + \eta + \frac{\ln(n)}{\eta T}$$

□

and if $T = \frac{4\ln(n)}{\epsilon^2}$ then:

$$v^* \leq \frac{1}{T}\sum_{t=1}^{T} A(p^t, j^t) \leq v^* + \epsilon$$

Now we show how to find an ϵ-optimal strategy for ROW. By the inequality above, there exists t such that $A(p^t, j^t) = \min_j A(p^t, j) \leq v^* + \epsilon$. Thus, if t minimizes $A(p^t, j^t)$ then p^t is an ϵ-optimal strategy for ROW. An ϵ-optimal strategy for COLUMN can also be found. Let q be such that $q_j = \frac{|\{t; j^t = j\}|}{T}$. For every i,

$$\frac{1}{T}\sum_{t=1}^{T} A(i, j^t) = A(i, q)$$

Therefore we get that:

$$v^* \leq \frac{1}{T}\sum_{t=1}^{T} A(p^t, j^t) \leq (1+\eta)\frac{1}{T}\sum_{t=1}^{T} A(i, j^t) + \frac{\ln n}{\eta T} \leq A(i, q) + \epsilon$$

Thus q is an ϵ-optimal strategy for column.

5. Maximum Multicommodity Flow

$G = (V, E)$ is a directed graph with n vertices and m edges. We are also given a capacity function $c : E \to \mathbb{R}^+$ and k pairs of source and sink. We want to maximize the total flow. Let $\mathcal{P}$ be the set of all simple paths from (s_i, t_i) for some $i \in [k]$. We show a $(1-\epsilon)$-approximation algorithm. We will use the **rewards** version of the Multiplicative Updates Algorithm. We will have an "expert" for each edge. Let $\eta = \frac{\epsilon}{2}$. We give each edge a weight w_e^t and initialize it to 1. In each iteration t, we find a shortest path p^t with respect to the edge weights $\frac{w_e^t}{c_e}$. We route c^t units of flow on the path p^t where $c^t = \min_{e \in p^t} c_e$.

Define $r_e^t = \frac{c^t}{c_e} \in [0, 1]$ if $e \in p^t$ and otherwise, $r_e^t = 0$. We stop when there is an edge $e \in E$ such that $\frac{f_e}{c_e} \geq \frac{\ln m}{\eta^2}$.

Analysis: Let f^{opt} be the optimal flow, and $F^{opt} = \sum_{p \in \mathcal{P}} f_p^{opt}$, Where f_p is the flow along the path p. Also let $F = \sum_{t=1}^{T} c^t$. By corrolary 1.2.2:

$$\sum_{t=1}^{T} p^t \cdot r^t \geq (1-\eta)\sum_{t=1}^{T} r_e^t - \frac{\ln m}{\eta}$$

It also holds that:

$$\sum_{t=1}^{T} p^t \cdot r^t = \sum_{t=1}^{T} \frac{\sum_{e \in p^t} w_e^t \frac{c^t}{c_e}}{\sum_{e \in E} w_e^t} = \sum_{t=1}^{T} c^t \frac{\sum_{e \in p^t} \frac{w_e^t}{c_e}}{\sum_{e \in E} w_e^t}$$

$$\leq \frac{\sum_{t=1}^{T} c^t}{F^{opt}} = \frac{F}{F^{opt}}$$

Let $p \in \mathcal{P}$ be a shortest path with respect to the edge weights $\frac{w_e}{c_e}$. We get that:

$$\frac{\sum_{e\in E} w_e}{\sum_{e\in p} \frac{w_e}{c_e}} \geq \frac{\sum_{e\in E} w_e \sum_{e\in p'} \frac{f_{p'}^{opt}}{c_e}}{\sum_{e\in p} \frac{w_e}{c_e}} = \frac{\sum_{p'\in\mathcal{P}} f_{p'}^{opt} \sum_{e\in p'} \frac{w_e}{c_e}}{\sum_{e\in p} \frac{w_e}{c_e}} \geq \sum_{p'\in\mathcal{P}} f_{p'}^{opt} = F^{opt}$$

Let $C = \max_{e\in E} \frac{f_e}{c_e}$. It follows that:

$$\frac{F}{F^{opt}} \geq \sum_t p^t r^t \geq (1-\eta) \max_{e\in E} \frac{f_e}{c_e} - \frac{\ln m}{\eta} \geq (1-2\eta)C$$

When the algorithm terminates $C \geq \frac{\ln m}{\eta^2}$, we scale down the flow by C and achieve a legal flow (a flow that satisfies the capacity constraints).

$$\frac{F}{C} \geq (1-2\eta)F^{opt} = (1-\epsilon)F^{opt}$$

5.1. **Bounding the number of iterations.** We stop when $C \geq \frac{\ln m}{\eta^2}$. Each iteration increases C by at least 1. Therefore, the number of iterations is bounded by $m\lceil \frac{\ln m}{\eta^2} \rceil$. Let $T_{sp}(m)$ be the time of finding a shortest path on a graph with $O(m)$ edges. Then the total running time is bounded by $O\left(k \frac{m \ln m}{\epsilon^2} T_{sp}(m)\right)$.