# 1 The Multiplicative Weights Update Method [1]

## 1.1 Setting

- At each time step $t = 1, \dots, T$:
  - We must decide for one of $m$ *experts*
  - Nature causes *event* $x_t$ to occur
  - Event $x$ incurs *penalty* $\mathbf{M}(e, x)$ to expert $e$. We suffer same penalty as chosen expert
- Goal: Difference between sum of our penalties and that of best expert should vanish

## 1.2 Multiplicative Weights Update Algorithm

- Input: Penalties $\mathbf{M}(e, x)$, number of rounds $T$, error parameter $\varepsilon \in (0, \frac{1}{2}]$

i) Initialize $w_{1,e} := 1$ for all $e$

ii) For $t := 1, \dots, T$:

   a) Randomly pick an expert according to distribution $\mathcal{D}_t$
   b) Observe nature choosing event $x_t$
   c) Multiplicative weights update: $w_{t+1,e} := w_{t,e} \cdot (1 - \varepsilon \mathbf{M}(e, x_t))$ for all $e$

**Table 1:** Symbols and Implicit Definitions

| Symbol / Definition | Meaning |
|---|---|
| $m$ | Number of experts |
| $\mathbf{M}(e, x)$ | Penalty for expert $e$ when event $x$ occurs |
| $w_{e,t}$ | Weight of expert $e$ at time step $t$ |
| $W_t := \sum_{e=1}^{m} w_{t,e}$ | Total weight of players at time step $t$ |
| $d_{t,e} := \frac{w_{t,e}}{W_t}$ | Probability of player $e$ being picked at time step $t$ |
| $\mathcal{D}_t := (d_{t,1}, \dots, d_{t,m})$ | Probability distribution for picking expert at time step $t$ |

## 1.3 Guarantees

- **Theorem**: Define $\mathbf{M}(\mathcal{D}_t, x_t) := \mathbf{E}_{e \sim \mathcal{D}_t}[\mathbf{M}(e, x_t)]$. Suppose all penalties $\mathbf{M}(e, x) \in [-1, 1]$. Then, for any expert $e$:

$$\sum_{t=1}^{T} \mathbf{M}(\mathcal{D}_t, x_t) \leq (1 + \varepsilon) \sum_{\geq 0} \mathbf{M}(e, x_t) + (1 - \varepsilon) \sum_{< 0} \mathbf{M}(e, x_t) + \frac{\ln m}{\varepsilon}$$

  Subscripts $\geq 0$ and $< 0$ refer to rounds $t$ with $\mathbf{M}(e, x_t) \geq 0$ and $< 0$ respectively. Remark: Also holds when $\mathbf{M}(e, x)$ contains *rewards* instead of penalties.

- Note first:

$$W_{T+1} = \sum_{e=1}^{m} w_{T,e} \cdot (1 - \varepsilon \mathbf{M}(e, x_T)) = W_T \cdot (1 - \varepsilon \mathbf{M}(\mathcal{D}_T, x_T)) = \dots = n \cdot \prod_{t=1}^{T} (1 - \varepsilon \mathbf{M}(\mathcal{D}_t, x_t))$$

- We show inequality after multiplying with $(-\varepsilon)$ and applying $\exp(\cdot)$:

$$
\begin{aligned}
\exp\Big(-\varepsilon \sum_{t=1}^{T} \mathbf{M}(\mathcal{D}_t, x_t)\Big) &\geq \prod_{t=1}^{T}(1 - \varepsilon \mathbf{M}(\mathcal{D}_t, x_t)) \qquad\qquad [\exp(x) \geq 1 + x] \\
&= \frac{W_{T+1}}{n} \geq \frac{w_{T+1,e}}{n} = \frac{1}{n}\prod_{t}(1 - \varepsilon \mathbf{M}(e, x_t)) \\
&\geq \frac{1}{n} \cdot (1-\varepsilon)^{\sum_{\geq 0} \mathbf{M}(e,x_t)} \cdot (1+\varepsilon)^{-\sum_{<0} \mathbf{M}(e,x_t)} \\
&\qquad \big[1 - \varepsilon x \geq (1-\varepsilon)^x \text{ for } x \in [0,1]\big] \\
&\qquad \big[1 - \varepsilon x \geq (1+\varepsilon)^{-x} \text{ for } x \in [-1,0]\big] \\
&= \exp(-\ln n) \cdot \exp\Big(\ln(1-\varepsilon) \sum_{\geq 0} \mathbf{M}(e, x_t)\Big) \\
&\quad \cdot \exp\Big(-\ln(1+\varepsilon) \sum_{<0} \mathbf{M}(e, x_t)\Big) \\
&\geq \exp\Big(-\varepsilon\Big[(1+\varepsilon)\sum_{\geq 0} \mathbf{M}(e, x_t) + (1-\varepsilon)\sum_{<0} \mathbf{M}(e, x_t) + \frac{\ln n}{\varepsilon}\Big]\Big) \\
&\qquad \big[\ln(1+x) \geq x(1-x) \text{ for } x \geq -0.68\ldots\big]
\end{aligned}
$$

- Assume from now on that all $\mathbf{M}(e, x) \in [-\ell, \varrho]$ where $0 \leq \ell \leq \varrho$.
- **Corollary**: The previous bound remains valid (simply scale all $\mathbf{M}(e, x)$ down by $\varrho$) except that the ln-term gets an additional factor $\varrho$.
- Note that

$$(1-\varepsilon)\sum_{<0} \mathbf{M}(e, x_t) \leq (1+\varepsilon)\sum_{<0} \mathbf{M}(e, x_t) + 2\varepsilon\ell T$$

  and

$$(1+\varepsilon)\frac{\sum_t \mathbf{M}(e, x_t)}{T} \leq \frac{\sum_t \mathbf{M}(e, x_t)}{T} + \varepsilon\varrho$$

- **Corollary**: Let $\delta > 0$. With $\varepsilon = \frac{\delta}{4\varrho}$ (w.l.o.g., $\varepsilon \leq \frac{1}{2}$), we have after $T = \frac{16\varrho^2 \ln n}{\delta^2}$ rounds, for every expert $e$:

$$\frac{\sum_t \mathbf{M}(\mathcal{D}_t, x_t)}{T} \leq \frac{\sum_t \mathbf{M}(e, x_t)}{T} + \underbrace{\frac{\varrho \ln n}{\varepsilon T} + 2\varepsilon\ell + \varepsilon\varrho}_{=\delta/4 + (\delta\ell)/(2\varrho) + \delta/4 \leq \delta}$$

  When $\ell = 0$, sufficient to choose $\varepsilon = \frac{\delta}{2\varrho}$ and $T = \frac{4\varrho^2 \ln n}{\delta^2}$.

### 1.4 Examples

- Kalai and Vempala [3]: Event $x_t = \mathit{cost}(\cdot, t) \in [0,1]^m$ and penalty $\mathbf{M}(e, x_t) = \mathit{cost}(e, t)$. Note: Equivalent guarantees as "follow the perturbed leader".
- Weighted-Majority Algorithm [4] is a deterministic variant: Event $x_t \in \{0, 1\}$ and $\mathbf{M}(e, x_t) = 1 - x_t$. Derandomized in that the majority gets all probability.

# 2 Applications

## 2.1 The Plotkin-Shmoys-Tardos Framework [5]

- Multiplicative weights method to determine approximate feasibility of linear program

$$Ax \geq (1 \dots 1)^T, \quad x \in P$$

  where $A \in \mathbb{R}^{m \times n}$ and $P \subseteq \mathbb{R}^n$ is a convex set. Note: Optimization via binary search.

- Input: Error parameter $\delta > 0$, $A$, $P$, parameters $0 \leq \ell \leq \varrho$, a subroutine
- Output: $x \in P$ with $A_e x \geq 1 - \delta$ for all rows $A_e \in \mathbb{R}^{1 \times n}$, or prove system infeasible
- Intuition: Expert = constraint, nature = subroutine, event = vector in $P$
- The subroutine ("oracle"):
  - Input: Probability distribution $\mathcal{D} = (d_1, \dots, d_m)$
  - Output: Vector $x \in P$ with $\sum_{e=1}^m d_e A_e x \geq 1$, or (correctly) say "infeasible"
  - For all points $x$ possibly returned assume for all constraints $e$ that $A_e x \in [-\ell, \varrho]$
- Adapt multiplicative weights update method as follows:
  - $\mathbf{M}(e, x) := A_e x$. Rationale: Reduce weight of well-satisfied constraints $\to$ similar in spirit to Langrangian relaxation
  - If subroutine ever returns "infeasible" return that system is infeasible: If $x$ with $Ax \geq (1 \dots 1)^T$ existed, then $\sum_{e=1}^m d_e A_e x \geq 1$ in particular.
  - Otherwise, return $x := \frac{1}{T} \sum_t x_t$. With $\varepsilon, T$ as in corollary, we have for all constraints $e'$:

$$\begin{aligned} 0 &\leq \frac{\sum_t \left(\sum_{e=1}^m d_{t,e} A_e x_t - 1\right)}{T} && \text{[all } x_t \text{ were feasible]} \\ &= \frac{\sum_t \mathbf{M}(\mathcal{D}_t, x_t)}{T} - 1 \leq \frac{\sum_t A_{e'} x_t}{T} + \delta - 1 = A_{e'} x - (1 - \delta) && \text{[Theorem]} \end{aligned}$$

## 2.2 Multicommodity Flow

- Maximum multicommodity flow (let $\mathcal{P}$ be union of all paths for every commodity):

$$\begin{aligned} \max \ & \sum_{p \in \mathcal{P}} x_p \\ \text{s.t. } & \frac{1}{c_e} \sum_{p \ni e} x_p \leq 1 \quad \forall e \in E \end{aligned}$$

- Want to find flow $x$ of volume $X$ (if feasible)
- Define modified edge costs at time step $t$ by $c'_{t,e} := \frac{d_{t,e}}{c_e}$
- Convex set $P = \{x \in \mathbb{R}^{\mathcal{P}}_{\geq 0} \mid \sum_{p \in \mathcal{P}} x_p = X\}$. Expert = constraint/edge. Subroutine: At time step $t$, finds flow (= event) $x \in P$ so that $\sum_e d_{t,e} \frac{1}{c_e} \sum_{p \ni e} x_p = \sum_p x_p \sum_{e \in p} c'_{t,e} \leq 1$
  $\to$ minimized by putting all flow on shortest path w.r.t. edge costs $c'_{t,e}$
- For $x \in P$, we can only guarantee $A_e x = \frac{1}{c_e} \sum_{p \ni e} x_p \leq \frac{X}{\min_e c_e} =: \varrho$. Hence, number of rounds $T$ only pseudo-polynomial.

### 2.3 The Garg-Könemann Algorithm for Multicommodity Flow [2]

- Expert = edge, event = shortest path $p_t$ w.r.t. edge costs $c'_{t,e}$
- Denote by $c^*_t$ the smallest capacity on path $p_t$. Rewards $\mathbf{M}(e, p_t) := \frac{c^*_t}{c_e}$ if $e \in p_t$ and 0 else
- Algorithm: In every round $t$, algorithm adds flow $c^*_t$ on path $p_t$. Suppose $f_e$ contains the current total amount of flow added to edge $e$. Then, the algorithm terminates after the round in which $\frac{f_e}{c_e} \geq \frac{\ln m}{\varepsilon^2}$ for some edge $e$. Return flow $(\max_e\{\frac{f_e}{c_e}\})^{-1} \cdot f$.
- Let $F = |f|$ total flow, $g$ some maximum flow, $G = |g|$. Immediately before return:

$$\begin{aligned}
\frac{F}{G} &= \sum_t \frac{c^*_t}{G} \geq \sum_t \frac{c^*_t \sum_{e \in p_t} c'_{t,e}}{\sum_{q \in \mathcal{P}} g_q \sum_{e \in q} c'_{t,e}} && [p_t \text{ is shortest path w.r.t. } c'_t] \\
&= \sum_t \frac{c^*_t \sum_{e \in p_t} c'_{t,e}}{\sum_e d_{t,e} \sum_{q \ni e} \frac{g_q}{c_e}} \geq \sum_t c^*_t \sum_{e \in p_t} c'_{t,e} && \Big[\textstyle\sum_{q \ni e} \frac{g_q}{c_e} \leq 1, \text{ for } g \text{ is feasible}\Big] \\
& && [\mathcal{D} \text{ is probability distribution}] \\
&= \sum_t \sum_{e \in E} d_{t,e} \mathbf{M}(e, p_t) \\
&= \sum_t \mathbf{M}(\mathcal{D}_t, p_t) \geq (1-\varepsilon) \max_e \Big\{ \sum_t \mathbf{M}(e, p_t) \Big\} - \frac{\ln m}{\varepsilon} && [\text{Theorem with signs reversed}] \\
&= (1-\varepsilon) \max_e \Big\{ \frac{f_e}{c_e} \Big\} - \frac{\ln m}{\varepsilon} \geq (1-2\varepsilon) \max_e \Big\{ \frac{f_e}{c_e} \Big\} && [\text{since } \tfrac{\ln m}{\varepsilon} \leq \varepsilon \max_e \big\{ \tfrac{f_e}{c_e} \big\}]
\end{aligned}$$

- Running time: Each edge can be minimum capacity edge at most $\lceil \frac{\ln m}{\varepsilon^2} \rceil$ times, therefore at most $m \cdot \lceil \frac{\ln m}{\varepsilon^2} \rceil$ iterations.

## References


[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta algorithm and its applications. Survey Paper, 2008. URL `http://www.cs.princeton.edu/~satyen/papers/MW_SDP.pdf`.

[2] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007. DOI: `10.1137/S0097539704446232`.

[3] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. DOI: `10.1016/j.jcss.2004.10.016`.

[4] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994. DOI: `10.1006/inco.1994.1009`.

[5] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995. DOI: `10.1287/moor.20.2.257`.