Some Applications:
- Learning
  - Online selection among experts
  - Boosting success of learning algorithms
    - e.g. Adaboost
- Optimization
  - Approximation algorithms for NP-hard problems
  - Solving semi-definite programs efficiently

Method has been used in many variants over the years
From a recent survey by Arora, Hazan, Kale:
  - This “meta algorithm and its analysis are simple and useful enough that they should be viewed as a basic tool taught to all algorithms students together with divide-and-conquer, dynamic programming, random sampling, and the like.”

Simple case: Stock market direction
- $n$ experts
- every day each expert $i$ makes a binary guess/prediction $g_i^{(t)}$ (up=$+1$ or down=$-1$)
- at end of the day can observe the outcome of what the market did that day: $o^{(t)}$
- After $T$ days, best expert $i^*$ gets return
  $$r_{i^*} = \max_i \sum_t o_i^{(t)} g_i^{(t)}$$
- The return $r_{i^*}=T-2m_{i^*}$ where $m_{i^*} = \#$ of mistakes in direction made by the best expert

Goal: Find a strategy that chooses an expert each day $t$ knowing only $o^{(s)}, g_i^{(s)}$ for $s<t$ and does not make many more mistakes than the best expert does
Warm-up: Weighted Majority Algorithm (Littlestone-Warmuth)

- Choose $\varepsilon \leq 1/2$
- Maintain a weight (confidence) in each expert $w_i$ and each day choose the prediction to be the weighted majority of their guesses; i.e. the sign of $\sum_j w_j g(t)_j$
  - Initially set each $w_i = 1$
  - No reason to prefer any expert
  - After each day replace $w_i$ by $w_i (1 - \varepsilon)$ if expert $i$ made a mistake
- Write $w(t)_i$ for value of $w_i$ at the start of $t$th day

Weighted Majority Algorithm

- Notation: $m_i(t) = \#$ of mistakes made by expert $i$ after $t$ steps
  $m(t) = \#$ of mistakes made by weighted majority after $t$ steps
- Theorem: For any expert $i$,
  $$m(T) \leq \frac{2}{\varepsilon} \ln n + 2(1+\varepsilon)m_i(T)$$

Weighted Majority Algorithm Proof

- Theorem: If $\varepsilon \leq 1/2$ then for any expert $i$,
  $$m(T) \leq \frac{2}{\varepsilon} \ln n + 2(1+\varepsilon)m_i(T)$$
- Proof:
  - Since each error accumulates a $(1 - \varepsilon)$ factor
    $$w(t+1)_i = (1 - \varepsilon) m_i(t)$$
  - Define “potential” = sum of expert weights:
    $$\Phi(t) = \sum_i w(t)_i$$
  - By definition $\Phi(t) = n$
  - Prediction is wrong only if at least $1/2$ the total weight of the experts is wrong
    - Potential will decrease by at least $\varepsilon \Phi(t)/2$
      - i.e., $\Phi(t+1) \leq (1 - \varepsilon/2) \Phi(t)$

Weighted Majority Algorithm Proof continued

- Theorem: For any expert $i$,
  $$m(T) \leq \frac{2}{\varepsilon} \ln n + 2(1+\varepsilon)m_i(T)$$
- Proof (continued):
  - $w(t+1)_i = (1 - \varepsilon) m_i(t)$
  - $\Phi(t+1) = n, \Phi(t+1) \leq (1 - \varepsilon/2) \Phi(t)$
  - So $\Phi(T+1) \leq n (1 - \varepsilon/2)^{m(T)}$
  - However $\Phi(T+1) \geq w(T+1)_i = (1 - \varepsilon) m_i(T)$ so
    $$n (1 - \varepsilon/2)^{m(T)} \geq (1 - \varepsilon) m_i(T)$$
  - Taking natural logarithms we get
    $$m(T) \ln (1 - \varepsilon/2) + \ln n \geq m_i(T) \ln (1-\varepsilon)$$
  - Theorem follows from $-x \geq \ln (1-x) \geq -x - x^2$ for $x \leq 1/2$
    - i.e. $m(T)(-\varepsilon/2) + \ln n \geq m_i(T) (-\varepsilon^2)$
More general experts scenario

- More general scenario:
  - \( n \) experts
  - every day each expert \( i \) chooses course of action
  - after it has been selected we find out that the \( i^{th} \) expert’s choice on day \( t \) incurs a cost \( m^{(t)}_i \) with \(-1 \leq m^{(t)}_i \leq 1\) (-ve cost implies a benefit)

- Goal: Find a (randomized) strategy of small expected total cost to choose course of action each day \( t \) knowing only \( m^{(s)}_i \) values for \( s < t \)

- In the simple case the costs \( m^{(t)}_i \) were
  - 0 (correct prediction) or 1 (mistake)

(Randomized) Multiplicative Weights Update Method

- Choose \( \epsilon \leq 1/2 \)
- Maintain a weight (confidence) in each expert \( w^{(t)}_i \) and each day choose course of action of \( i^{th} \) expert with probability proportional to its current weight; i.e. with prob \( p^{(t)}_i = w^{(t)}_i / \sum_j w^{(t)}_j \)
- Set each \( w^{(t)}_i = 1 \)
  - No reason to prefer any expert at start
- Set \( w^{(t+1)}_i = w^{(t)}_i (1 - \epsilon m^{(t)}_i) \)

- Define \( \Phi^{(t)} = \sum_j w^{(t)}_j \) as before so \( p^{(t)}_i = w^{(t)}_i / \Phi^{(t)} \)

- Note: Average behavior similar to weighted majority for binary predictions (bias of \( t^{th} \) prediction is the average prediction, not its sign)

Multiplicative Weights Update Method

- Expected cost of choice in the \( t^{th} \) step is \( M_i^{(t)} = \sum_i p^{(t)}_i m^{(t)}_i = \sum_i w^{(t)}_i m^{(t)}_i / \Phi^{(t)} \)

- Notation:
  \( M_i^{(t)} = \sum_{s \leq t} m^{(s)}_i \) = total cost for expert \( i \) in first \( t \) steps
  \( M^{(t)} = \sum_{s \leq t} M_s \) = expect total cost of multiplicative update choices in first \( t \) steps

- Theorem: For any expert \( i \),
  \( M^{(T)} \leq (1/\epsilon) \ln n + M_i^{(T)} + \epsilon \sum_{t \leq T} |m^{(t)}_i| \)

  Proof:
  - Now \( \Phi^{(t+1)} = \sum_i w^{(t+1)}_i \)
    \[ = \sum_i w^{(t)}_i (1 - \epsilon m^{(t)}_i) \]
    \[ = \Phi^{(t)} - \epsilon \sum_i p^{(t)}_i \Phi^{(t)} m^{(t)}_i \] since \( p^{(t)}_i = w^{(t)}_i / \Phi^{(t)} \)
    \[ = \Phi^{(t)} (1 - \epsilon \sum_i p^{(t)}_i m^{(t)}_i) = \Phi^{(t)} (1 - \epsilon M_i^{(t)}) \]
    \[ \leq \Phi^{(t)} e^{-\epsilon M_i^{(t)}} \] since \( 1 + x \leq e^x \)

  - By definition \( \Phi^{(1)} = n \) so
    \( \Phi^{(T+1)} \leq n e^{-\epsilon (M_1 + \ldots + M_T)} = n e^{-\epsilon M^{(T)}} \)
Multiplicative Weights Update Method

Theorem: If $\varepsilon \leq \frac{1}{2}$ then for any expert $i$,

$$M(T) \leq \left(\frac{1}{\varepsilon}\right) \ln n + M_i(T) + \varepsilon \sum_{t \leq T} |m^{(t)}_i|$$

Proof (continued):

$$\Phi^{T+1} \leq n e^{-\varepsilon M(T)}$$

But

$$\Phi^{T+1} \geq w^{(T+1)}_i = (1-\varepsilon m^{(1)}_i)(1-\varepsilon m^{(2)}_i)\ldots(1-\varepsilon m^{(T)}_i)$$

Taking natural logarithms we get

$$-\varepsilon M(T) + \ln n \geq \sum_{t \leq T} \ln (1-\varepsilon m^{(t)}_i)$$

Theorem follows from $\ln (1-x) \geq -x-x^2$ and $\ln (1+x) \geq x-x^2$ for $0 \leq x \leq \frac{1}{2}$

Simple Application: Approximating Minimum Set Cover

Minimum-Set-Cover:

- Given a universe $U=\{1,\ldots,n\}$, a collection $S_1,\ldots,S_m$ of subsets of $U$ find a minimum number $OPT$ of sets in the collection that covers every element of $U$.

Where are the experts?

- Each element $i$ of $U$ will be an expert

What are the time steps?

- Each time step $t$ will correspond to a set $S_t$

What are the costs?

- $m^{(t)}_i=1$ if $i \in S_t$ and $=0$ if not

What do the weights look like?

- Set $\varepsilon=1$ (will use even simpler analysis here)
- Now $w^{(1)}_i=1$ and $w^{(t+1)}_i = w^{(t)}_i (1-\varepsilon m^{(t)}_i)$ so $w^{(t+1)}_i=0$ iff $i$ is contained in $S_1 \cup \ldots \cup S_t$

We will have an adversary order the sets:

- At step $t$ the adversary will choose the set $S_t$ that has the most uncovered elements (Greedy choice)
  - the set maximizing $\sum_{i \in S_t} p^{(t)}_i = \sum_{i \in S_t} w^{(t)}_i / \Phi^{(t)}$
Simple Application: Approximating Minimum Set Cover

- Adversary makes Greedy choice of set $S_{jt}$ maximizing $\sum_{i \in S_{jt}} p^{(t)}_i$
  - Now $p^{(t)}_1, \ldots, p^{(t)}_n$ is a probability distribution on elements
  - Since OPT sets are enough to cover all elements, there must exist some set $S_{jt}$ with
    $$1/OPT \leq \sum_{i \in S_{jt}} p^{(t)}_i = \sum_{i \in S_{jt}} w^{(t)}_i / \Phi^{(t)}$$
  - So $\sum_{i \in S_{jt}} w^{(t)}_i \geq \Phi^{(t)} / OPT$
  - and $\Phi^{(t+1)} = \Phi^{(t)} - \epsilon \sum_{i \in S_{jt}} w^{(t)}_i \leq \Phi^{(t)} (1 - 1/OPT)$
  - It follows that $\Phi^{(t+1)} < n e^{t/OPT}$

- It follows that $\Phi^{(t+1)} < n e^{t/OPT}$

- Now $\Phi^{(t+1)}$ is just the total # of uncovered elements after choice of first $t$ sets
  - When $t/OPT \geq \ln n$ we have $\Phi^{(t+1)} < n e^{\ln n} = 1$ and every element must be covered by the adversary’s choice of sets so far

- This says that the Greedy algorithm (the adversary’s strategy) will produce a set cover of size at most $\lceil \ln n \rceil \cdot OPT$
  - This is essentially the best possible approximation factor unless $P=NP$