CSE 421: Introduction to Algorithms

Multiplicative Weights Update Method

Paul Beame

Multiplicative Weights Update Method

- 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

Multiplicative Weights Update Method

- 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."
  - http://www.cs.princeton.edu/~satyen/papers/mw-survey.pdf

Online choice from experts

- Simple case: Stock market direction
  - **n** experts
  - every day each expert **i** makes a binary guess/prediction $g^{(t)}_i$ (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

    $$r_{i^*} = \max_i \sum_t o^{(t)} g^{(t)}_i$$
  - 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 $o^{(s)}$, $g^{(s)}_i$ for $s<t$ and does not make many more mistakes than the best expert does

Warm-up: Weighted Majority Algorithm (Littlestone-Warmuth)

- Choose $\varepsilon \le 1/2$
- 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 $\sum_j w_j g^{(t)}_i$
  - 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,
  $m(T) \le (2/\varepsilon)\ \ln n + 2(1+\varepsilon)m_i(T)$

Weighted Majority Algorithm Proof

- Theorem: If $\varepsilon \le ½$ then for any expert i,
  $m(T) \le (2/\varepsilon)\ \ln n + 2(1+\varepsilon)m_i(T)$
- 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 $\Phi^{(1)} = n$
  - Prediction is wrong only if at least ½ the total weight of the experts is wrong
    - Potential will decrease by at least $\varepsilon\ \Phi^{(t)}/2$
    - i.e., $\Phi^{(t+1)} \le (1 - \varepsilon/2)\ \Phi^{(t)}$

Weighted Majority Algorithm Proof continued

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

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 $-1 \le m^{(t)}_i \le 1$ (-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 $\varepsilon \le 1/2$
- 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 $p^{(t)}_i = w^{(t)}_i / \Sigma_j w^{(t)}_j$
  - Set each $w^{(1)}_i = 1$
    - No reason to prefer any expert at start
  - Set $w^{(t+1)}_i = w^{(t)}_i (1 - \varepsilon\, m^{(t)}_i)$
- Define $\Phi^{(t)} = \Sigma_j w^{(t)}_i$ 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
  $M_t = \Sigma_i p^{(t)}_i m^{(t)}_i = \Sigma_i w^{(t)}_i m^{(t)}_i / \Phi^{(t)}$
- Notation:

  $M_i(t) = \Sigma_{s\le t} m^{(s)}_i$ = total cost for expert i in first t steps

  $M(t) = \Sigma_{s\le t} M_s$ =expect total cost of multiplicative update choices in first t steps
- Theorem: For any expert i,

  $$M(T) \le (1/\varepsilon) \ln n + M_i(T) + \varepsilon\, \Sigma_{t\le T} |m^{(t)}_i|$$

Multiplicative Weights Update Method

- Theorem: If $\varepsilon \le ½$ then for any expert i,

  $$M(T) \le (1/\varepsilon) \ln n + M_i(T) + \varepsilon\, \Sigma_{t\le T} |m^{(t)}_i|$$
- Proof:
  - Now $\Phi^{(t+1)} = \Sigma_i w^{(t+1)}_i$

    $= \Sigma_i w^{(t)}_i (1 - \varepsilon\, m^{(t)}_i)$

    $= \Phi^{(t)} - \varepsilon\, \Sigma_i p^{(t)}_i \Phi^{(t)} m^{(t)}_i$ since $p^{(t)}_i = w^{(t)}_i / \Phi^{(t)}$

    $= \Phi^{(t)}(1 - \varepsilon\, \Sigma_i p^{(t)}_i m^{(t)}_i) = \Phi^{(t)} (1 - \varepsilon M_t)$

    $\le \Phi^{(t)} e^{-\varepsilon M_t}$ since $1+x \le e^x$
  - By definition $\Phi^{(1)} = n$ so
    $\Phi^{(T+1)} \le n\, e^{-\varepsilon (M_1 + \ldots + M_T)} = n\, e^{-\varepsilon M(T)}$

Multiplicative Weights Update Method

- Theorem: If $\varepsilon \le ½$ then for any expert i,

  $$M(T) \le (1/\varepsilon) \ln n + M_i(T) + \varepsilon \sum_{t \le T} |m^{(t)}_i|$$

- Proof (continued):
  - $\Phi^{(T+1)} \le n\, e^{-\varepsilon M(T)}$
  - But $\Phi^{(T+1)} \ge w^{(T+1)}_i$

    $= (1-\varepsilon\, m^{(1)}_i)\,(1-\varepsilon\, m^{(2)}_i) \ldots (1-\varepsilon\, m^{(T)}_i)$
  - Taking natural logarithms we get

    $-\varepsilon M(T) + \ln n \ge \sum_{t \le T} \ln (1-\varepsilon\, m^{(t)}_i)$
  - Theorem follows from $\ln(1-x) \ge -x-x^2$ and $\ln(1+x) \ge x-x^2$ for $0 \le x \le ½$ ■

Multiplicative Weights Update Method

Corollary: If $\varepsilon \le ½$ and all costs are positive then for any expert i,

$$M(T) \le (1/\varepsilon) \ln n + (1+\varepsilon)\, M_i(T)$$

Note: The same holds if $M_i(T)$ is replaced by the cost of the best fixed random distribution of experts since one might just as well pick the best one.

The guarantee holds even if an adversary gets to choose the costs at time t after seeing the entire run of the algorithm up to time t

Lots of variants when there is a cost to change experts or one obtains only partial information about outcomes

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_{j_t}

Simple Application: Approximating Minimum Set Cover

What are the costs?
- $m^{(t)}_i=1$ if $i \in S_{j_t}$ 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 $S_{j_1} \cup \ldots \cup S_{j_t}$

We will have an adversary order the sets:
- At step t the adversary will choose the set S_{j_t} that has the most uncovered elements (Greedy choice)
  - the set maximizing $\sum_{i \in S_{j_t}} p^{(t)}_i = \sum_{i \in S_{j_t}} w^{(t)}_i / \Phi^{(t)}$

Simple Application: Approximating Minimum Set Cover

- Adversary makes Greedy choice of set S_{j_t} maximizing $\sum_{i\in S_{j_t}} p^{(t)}_i$
  - 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_{j_t} with

    $$1/OPT \le \sum_{i\in S_{j_t}} p^{(t)}_i = \sum_{i\in S_{j_t}} w^{(t)}_i \,/\, \Phi^{(t)}$$
  - So $\sum_{i\in S_{j_t}} w^{(t)}_i \ge \Phi^{(t)} /OPT$

    and $\Phi^{(t+1)} = \Phi^{(t)} - \varepsilon \sum_{i\in S_{j_t}} w^{(t)}_i \le \Phi^{(t)} (1 - 1/OPT)$

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

Simple Application: Approximating Minimum Set Cover

- 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 $t/OPT \ge \ln n$ 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**