CS 224: Advanced Algorithms Spring 2017

Lecture 23 — April 11, 2017

Prof. Jelani Nelson *Scribe: Thomas Orton*

1 Overview

Today we'll cover the Multiplicative Weights (MW) algorithm and begin showing how it can be applied to approximately solve LPs. Recall the setup of the expert prediction problem from the previous lecture: We have n experts, and we make binary predictions every day for T days. Each day $t \in [1, T]$ has a correct answer $a_t \in \{0, 1\}$ which we would like to guess. During each day t, each of the n experts makes a prediction of what a_t is. Given each expert's prediction, and $a_0, ..., a_{t-1}$, we would like to come up with a rule for guessing a_t such that we don't do much worse than if we had known $a_1, ..., a_T$ in advance, and had selected a single expert to copy for each of the T days (an expert which achieves minimum error on $a_1, ..., a_T$).

1.1 Review from last lecture

Some notation: Let M^t denote the number of mistakes made by our prediction algorithm between times 1 and t, inclusive, and let m_i^t be the number of mistakes by expert i in the same time period.

Theorem 1. *(From last lecture): There exists an algorithm s.t.*

$$M^T \leqslant \left(\lceil \log_2(n) \rceil \cdot \min_{1 \leqslant i \leqslant n} m_i^T \right) + \lceil \log_2(n) \rceil$$

This algorithm worked by taking the majority vote of all "living" experts, and killing experts when they made a prediction mistake. When all the experts were no longer living, they are all revived.

2 The MW algorithm

The MW algorithm has been reinvented many times in different fields (also known as the Hedge algorithm). For a survey, see [1]. The MW algorithm can be loosely seen as a generalization of the previous method we used for expert prediction: beforehand, we really took a "weighted" majority vote to be our prediction, where experts which were not alive had weight 0, and experts which were living had weight 1. Instead of setting the weights of incorrect experts to 0, MW decreases them gradually by a constant factor.

2.1 Algorithm

We use $[z]$ to denote $\{1, \ldots, z\}$ for a positive integer z.

- We are given a parameter η.
- Let w_i^t be the weight of expert i at time $t \in [T+1]$.
- Initialize expert weights $w_i^1 = 1$ for each $i \in [n]$.
- On day t, we make the prediction $b_t \in \{0, 1\}$ such that b_t is the prediction with the greatest weight. Specifically, if S_1 (S_0) is the set of the indices of experts which predict 1 (0) at time t, then we choose b_t such that $\sum_{i \in S_{p_t}} w_i^t \geqslant \sum_{i \in S_{(1-p_t)}} w_i^t$.
- Let W^t be the set of indicies of experts which made an incorrect prediction at time t. We set $\forall i \in W^t$, $w_i^{t+1} \leftarrow (1-\eta) w_i^t$, and $\forall i \in (\{1, ..., n\} - W_t)$, $w_i^{t+1} \leftarrow w_i^t$.

2.2 Performance

Theorem 2. *For $\eta \in (0, \frac{1}{10}]$, MW_η achieves error bounds*

$$M^T \leqslant (2+\eta) \min_{1 \leqslant i \leqslant n} m_i^T + \frac{2\ln(n)}{\eta}$$

Proof. We'll prove this theorem by defining a "potential" $\Phi^t = \sum_{i=1}^n w_i^t$, and giving lower and upper bounds on this potential in terms of quantities we are interested in. We'll complete the proof by relating these bounds and solving for M^T (this can be seen as general proof technique for certain classes of problems).

First, suppose our algorithm makes a mistake on day t. This means that at least half of the total weight at time t is placed on experts which predicted incorrectly at time t. For these incorrect experts, their weight decreases by a factor of $(1-\eta)$. The remaining experts have their weights unchanged. We therefore have

$$\Phi^{t+1} \leqslant (1-\eta) \times \frac{1}{2}\Phi^t + \frac{1}{2}\Phi^t = (1 - \frac{\eta}{2})\Phi^t$$

By induction, and noting that $\Phi^1 = n$, we get

$$\Phi^{T+1} \leqslant n \cdot (1 - \frac{\eta}{2})^{M^T}$$

This gives us our upper bound in terms of M^T. To get a lowerbound, note that Φ^{T+1} is at least as large as the weight of any particular expert at time $T+1$. We therefore get

$$\forall i \in [n],\ \Phi^{T+1} \geqslant w_i^{T+1} = (1-\eta)^{m_i^T}$$

Combining these bounds gives

$$(1-\eta)^{m_i^T} \leqslant n(1-\frac{\eta}{2})^{M^T}$$
$$m_i^T \ln(1-\eta) \leqslant \ln(n) + M^T \ln(1-\frac{\eta}{2})$$

Now recall Taylor's theorem: Since $f(x) := \ln(1-x)$ is twice differentiable in $[0, \frac{1}{10}]$ we can write $\forall x \in [0, \frac{1}{10}]$, $f(x) = f(0) + xf'(0) + x^2 f''(z_x)/2$ for $z_x \in [0, x]$. One can use this to show $-x - x^2 \leqslant \ln(1-x) \leqslant -x$ for any $x \in [0, \frac{1}{10}]$. Notice that this is the part of the proof that requires η to be a sufficiently small constant. Using this bound, we can write $\forall i \in [n]$

$$m_i^T(-\eta - \eta^2) \leqslant m_i^T \ln(1-\eta) \leqslant \ln(n) + M^T \ln(1-\frac{\eta}{2}) \leqslant \ln(n) - \frac{\eta}{2} M^T$$

$$M^T \leqslant \left[(\eta + \eta^2) m_i^T + \ln(n)\right] \times \frac{2}{\eta} = (2+\eta) m_i^{(T)} + \frac{2\ln(n)}{\eta}$$

Since this is true for all $i \in [n]$, we are free to pick i to be the index of the expert which achieves minimum error.

□

3 A Slightly Different Expert Prediction Problem

Let's consider a modified expert prediction task to solve. Instead of giving a prediction every day, we need to give a probability distribution over experts every day. For each day $t \in [T]$, after choosing a probability distribution $p^t \in \Delta^n$ over the experts, we receive a cost vector $m^t \in [-1, 1]^n$, where we write $m^t = (m_1^t, ..., m_n^t)$. Think of m_i^t as being the cost of choosing expert i at time t, where the more negative a cost is the better. We want to come come up with a strategy for choosing our distributions $p^1, ..., p^T$ such that $\sum_{t=1}^T \langle p^t, m^t \rangle$ is small, where we think of $\langle p^t, m^t \rangle$ as being the expected cost of choosing experts from distribution p^t at time t. Now we can imagine, say, that each expert is a stock, and m_i^t is the number of cents we lose per dollar invested in stock i on day t. Then p^t essentially decides our portfolio on day t (making the simplifiying assumption that there are no transaction costs in buying/selling stocks).

To deal with this problem, we can modify the weight update rule for MW to depend on the cost of the experts at each day t. Specifically, we now update weights as $w_i^{t+1} \leftarrow (1 - \eta m_i^t) w_i^t$. On day t, we will pick the "weighted expert" p^t such that $p_i^t := w_i^t / \Phi^t$, where $\Phi^t := \sum_{i=1}^n w_i^t$ as before.

3.1 Performance

Theorem 3. *$\forall \eta \in (0, \frac{1}{10}]$, we have*

$$\sum_{t=1}^{T} \langle p^t, m^t \rangle \leqslant \min_i \sum_{t=1}^{T} m_i^t + \eta(\sum_{t=1}^{T} |m_i^t|) + \frac{\ln n}{\eta}$$

Comment: Note that if we let OPT denote the minimum cost achievable by picking a single fixed expert to have probability 1 for all $t \in T$, then the above bound is

$$\leqslant OPT + \eta T + \ln(n)/\eta \leqslant OPT + 2\sqrt{T\ln(n)}$$

By choosing $\eta = \sqrt{\ln(n)/T}$ for sufficiently large T. The $2\sqrt{T\ln(n)}$ term is usually called the "regret" achieved by the algorithm. Often bounds will be divided by T on both sides to get a bound on the "average regret per day", which in this case goes to 0 as $T \to \infty$ as $O((\ln n)/\sqrt{T})$ for the same choice of η.

Proof. We'll use the same bounding technique as before. Recall the identity $p_i^t := w_i^t/\Phi^t$. We have

$$\Phi^{t+1} = \sum_{i=1}^{n} w_i^{t+1} = \sum_{i=1}^{n}(1-\eta m_i^t)w_i^t = \Phi^t - \eta\sum_{i=1}^{n} m_i^t w_i^t = \Phi^t - \eta\sum_{i=1}^{n} m_i^t(p_i^t\Phi^t)$$

$$= (1-\eta\langle m^t, p^t\rangle)\Phi^t \leqslant e^{-\eta(m^t.p^t)}\Phi^t$$

since $1 + x \leqslant e^x$. By induction, and noting that $\Phi^1 = n$, we get

$$\Phi^{T+1} \leqslant ne^{-\eta\sum_{t=1}^{T}\langle m^t, p^t\rangle}$$

Now we want to get a lower bound on Φ^{T+1}. $\forall i \in [n]$, we have $\phi^{T+1} \geqslant w_i^{T+1} = \prod_{t=1}^{T}(1-\eta m_i^t)$. We can write

$$\prod_{t=1}^{T}(1-\eta m_i^t) = \prod_{\geqslant 0}(1-\eta m_i^t)\prod_{<0}(1-\eta m_i^t)$$

where "$\geqslant 0$"$= \{t \in [1,T] | m_i^t \geqslant 0\}$ and "< 0"$= \{t \in [1,T] | m_i^t < 0\}$.

Aside: if $x \in [0,1]$ and $\epsilon \in (0,1)$, we have $(1-\epsilon)^x \leqslant (1-\epsilon x)$, and if $x \in [-1,0]$ and $\epsilon \in (0,1)$, we have $(1+\epsilon)^{-x} \leqslant (1-\epsilon x)$. For example, to prove the first bound we can notice that $f(x) := (1-\epsilon)^x$ is convex. For convex functions f, we have that $\forall t \in [0,1]$, $f(tx_1 + (1-t)x_2) \leqslant tf(x_1) + (1-t)f(x_2)$. Thus we have $f(x \times 1 + (1-x) \times 0) \leqslant xf(1) + (1-x)f(0) = 1 - \epsilon x$.

Using the aside, by setting $x = m_i^t$ and $\epsilon = \eta$, we can get the inequalities

$$\prod_{\geqslant 0}(1-\eta m_i^t) \geqslant (1-\eta)^{\sum_{\geqslant 0} m_i^t}$$

$$\prod_{<0}(1-\eta m_i^t) \geqslant (1+\eta)^{\sum_{<0} -m_i^t}$$

Putting the upper and lower bounds together, we get:

$$(1-\eta)^{\sum_{\geqslant 0} m_i^t}(1+\eta)^{\sum_{<0} -m_i^t} \leqslant n e^{-\eta \sum_{t=1}^{T}\langle m^t, p^t\rangle}$$

Thus

$$-\left(\sum_{\geqslant 0} m_i^t\right)\ln(1-\eta)+\left(\sum_{<0} m_i^t\right)\ln(1+\eta)+\ln n \geqslant \eta \sum_{t=1}^{T} m^t p^t$$

Using the previous bound $-\eta-\eta^2 \leqslant \ln(1-\eta)$ for $\eta \in [0, \frac{1}{10}]$, and another bound on $(\eta-\eta^2) \leqslant \ln(1+\eta)$ (also derived from Taylor's theorem), we get

$$\sum_{t=1}^{T}\langle m^t, p^t\rangle \leqslant \frac{1}{\eta}\left(\sum_{\geqslant 0} m_i^t\right)(\eta+\eta^2)+\frac{1}{\eta}\left(\sum_{<0} m_i^t\right)(\eta-\eta^2)+\frac{\ln n}{\eta}$$

$$=\sum_{t=1}^{T} m_i^t+\eta\sum_{t=1}^{T}|m_i^t|+\frac{\ln n}{\eta}$$

Picking the $i \in [n]$ which minimizes this expression completes the proof.

□

4 Applying MW to LP: Overview and Example

The approach which follows appeared in [2].

Problem ⊕ : We are given a matrix A, a vector b, and convex region P. If $\{x \in P, Ax \geqslant b\}$ is feasible, we want to find an x s.t. $x \in P$ and $Ax \geqslant b - \epsilon\vec{1}$ (otherwise we should return "error").

Example of an application: Recall setting up the fractional set cover problem as an LP. Using the notation which appeared in previous lectures, we can define this LP by writing

$$\min \sum_{S} x_S \qquad \text{s.t. } \forall e \qquad \sum_{S \ni e} x_S \geqslant 1 \qquad \text{and} \qquad x \geqslant 0$$

This definition implicitly defines A, b in the LP. Now imagine that we remove the min objective by replacing it with the constraint $\sum_S x_S \leqslant \beta$, where β is a parameter we will perform a binary search on. Now we define $P = \{x : x \geqslant 0, \sum_S x_S \leqslant \beta\}$, and we pass P, A, b to an algorithm which solves ⊕. Assuming we've chosen a β such that the problem has a feasible solution, we'll get back an $x \in [0,1]^m$ such that $\forall e$, $\sum_{S \ni e} X_S \geqslant 1-\epsilon$, $x \geqslant 0$, and $\sum_S x_S \leqslant \beta$. If we set $\tilde{x} = x/(1-\epsilon)$, then $\tilde{x}$ is a feasible solution to our original problem, and $\sum_S \tilde{x}_S \leqslant (1+O(\epsilon))\sum_S x_S$. Assuming we can solve ⊕, we have therefore given a procedure which will give us a $(1+O(\epsilon))OPT$ approximation for our fractional set cover LP.

Question: How does MW fit into all of this? Consider the following definition and theorem.

Definition 4. *Given P, A, b, a "γ bounded oracle" is an algorithm which takes as input any probability distribution q over the rows of A (interpreted as a vector in the simplex). If $((x \in P) \wedge (q^T Ax \geqslant q^T b))$ is feasible, then the oracle returns an $x^* \in P$ such that for every row index i of A, $|\langle A_i, x^* \rangle - b_i| \leqslant \gamma$. If $(x \in P \wedge q^T Ax \geqslant q^T b)$ is not feasible, the oracle outputs "infeasible".*

Theorem 5. *There is a way to run MW_η, using a γ bounded oracle as a subroutine, to solve $\oplus$. In this construction, we set $\eta = \frac{\epsilon}{2\gamma}$ and $T = \frac{4\gamma^2 \ln(m)}{\epsilon^2}$ (to be seen next lecture).*

To conclude, we'll briefly give a way to construct a γ bounded oracle for the fractional set cover problem. To start off, we want to know whether $(x \geqslant 0 \wedge \sum x_S \leqslant \beta \wedge q^T Ax \geqslant q^T b)$ is feasible. For this particular problem, we know $q^T b = 1$ and $q^T Ax = \sum_{e=1}^n q_e (\sum_{S \ni e} x_S) = \sum_S x_S q(S)$ where $q(S) := \sum_{e \in S} q_e$. Our constraints therefore require that $\sum_S x_S q(S) \geqslant 1$. Given q, want to find an x which satisfies this inequality. Notice that we can maximize $\sum_S x_S q(S)$ subject to the constraints $x \geqslant 0$ and $\sum x_S \leqslant \beta$ as follows: put all β of x's mass on the entry where q is largest, i.e. $x_{S*} = \beta$, $x_S = 0$ for $S \neq S^*$, where $S^* = \arg\max q(S)$. If this choice of x does not satisfy the inequality, output "infeasible" (we can't do any better, so the system isn't feasible). Otherwise, return x.

It remains to determine the γ we get when the inequality holds, i.e. we want to ideally find a small γ such that $|\langle A_e, x \rangle - 1| \leqslant \gamma$ holds. There are two cases to consider. First suppose $e \notin S^*$. Then by our choice of x we have $|\langle A_e, x \rangle - 1| \leqslant 1$. If $e \in S^*$, then $|\langle A_e, x \rangle - 1| = |\beta - 1| = \beta - 1$ (because $\beta \geqslant 1$ for this problem to have a solution). Thus we can set $\gamma = max(1, \beta - 1) \leqslant \beta \leqslant m$, where m is the number of sets.

References

[1] Sanjeev Arora, Elad Hazan, Satyen Kale. The Multiplicative Weights Update Method: A Meta-Algorithm and Applications *Theory of Computing*, 8(1): 121-164 2012.

[2] Serge A. Plotkin, David B. Shmoys, Éva Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Math. Oper. Res.*, 20(2): 257-301 1995.