Machine Learning Theory
Lecture 16

Nicholas Harvey

November 2, 2018

1 The randomized weighted majority algorithm

Today we will show that the unwanted factor of 2 can be removed from the Weighted Majority algorithm, through the use of randomization. Or rather, we weaken the adversary that decides the costs from being an *adaptive offline adversary* (who knows the algorithm's random bits) to being an *adaptive online adversary* (who knows the algorithm but not its random bits).

We will show that the algorithm's *expected* cost can be made arbitrarily close to C_{j^*} as $\eta \to 0$ and $T \to \infty$. Moreover, the algorithm can achieve this guarantee without receiving any predictions! In the present scenario, expert j incurs an arbitrary cost $c_{i,j} \in [0,1]$ in each time step i, and the algorithm must randomly choose an expert to follow without any indication of what the expert's cost will be.

Algorithm 1 The randomized weighted majority algorithm.

```
procedure RandomizedWeightedMajority($\eta$)
    Let $y_1 = (1, ..., 1)$
    for $i \leftarrow 1, 2, ...$ do
        Let $x_i = y_i / \|y_i\|_1$ (normalize the weights).
        Follow expert $j$ with probability $x_{i,j}$.
        ▷ Expected cost incurred is $\sum_{j=1}^{n} c_{i,j} x_{i,j} = c_i^\mathsf{T} x_i$
        ▷ Receive cost vector $c_i$
        for $j \leftarrow 1, ..., n$ do
            $y_{i+1,j} = y_{i,j} \exp(-\eta c_{i,j})$        (decrease weight according to expert $j$'s cost)
```

Theorem 1.1. Assume that the costs satisfy $c_{i,j} \in [0,1]$ for all i, j. Let $\epsilon = 1 - e^{-\eta} \approx \eta - \eta^2/2$ (so $\eta = \ln \frac{1}{1-\epsilon} \approx \epsilon + \epsilon^2/2$). Assume that $\epsilon \leq 1/2$. Consider any time step t. Let A be the total expected cost of the algorithm at time t. Let j^* be the expert with minimum total cost. Then

$$A \;\leq\; (1+\epsilon) \sum_{i=1}^{t} c_{i,j^*} + \frac{\ln n}{\epsilon}.$$

References. [5, Lemma 1.4], [6, Theorem 21.11].

Note. The costs c_i are determined by an ***adaptive online adversary***: they can depend on the algorithm's choice of experts in rounds $1, ..., i-1$ (and therefore also can depend on the previous costs $c_1, ..., c_{i-1}$). However, they *cannot* depend on the expert j chosen by the algorithm in round i, as j is determined using the algorithm's private randomness. Note that c_i can depend on the *distribution* x_i, which is completely determined by the previous costs $c_1, ..., c_{i-1}$.

Remark. Note that the bound gives a multiplicative guarantee with respect to the cost of the best expert, and a small $O(\log n)$ additive term.

Proof Idea. As in the proof last time of Weighted Majority, it is useful to use $\|y_i\|_1$ as a potential function. Step (b) is unchanged: the total cost of expert j^* is bounded using the potential. Step (a) is a bit different: we will relate the change in potential $\|y_{i+1}\|_1 \,/\, \|y_i\|_1$ to the *expected* cost incurred by the algorithm at time step i. In doing so, we will avoid the unwanted factor 2 that arose with Weighted Majority.

Proof.

Step (a):

$$\frac{\|y_{i+1}\|_1}{\|y_i\|_1} \;=\; \sum_{j=1}^{n} \frac{y_{i,j}\exp(-\eta c_{i,j})}{\|y_i\|_1} \;\leq\; \sum_{j=1}^{n} x_{i,j}(1-\epsilon c_{i,j}) \;=\; 1-\epsilon\sum_{j=1}^{n} x_{i,j}c_{i,j} \;\leq\; \exp(-\epsilon x_i^{\mathsf{T}} c_i).$$

The first inequality uses Claim 3.2 with $\alpha = e^{-\eta}$ and the definition $\epsilon = 1-e^{-\eta}$. The second inequality is Claim 3.1. Thus, taking the product for $i = 1, ..., t$, we see that the total weight at step $t+1$ is related to the total cost incurred by the algorithm:

$$\|y_{t+1}\|_1 \;\leq\; \underbrace{\|y_1\|_1}_{=n} \cdot \exp\Big(-\epsilon \underbrace{\sum_{i=1}^{t} x_i^{\mathsf{T}} c_i}_{\substack{\text{total expected}\\ \text{cost of algorithm}}}\Big) \;=\; n\exp(-\epsilon A). \tag{1.1}$$

Step (b): As before,

$$\underbrace{y_{1,j^*}}_{=1} \cdot \exp\Big(-\eta \underbrace{\sum_{i=1}^{t} c_{i,j^*}}_{\substack{\text{total cost of}\\ \text{expert } j^*}}\Big) \;=\; y_{t+1,j^*} \;\leq\; \|y_{t+1}\|_1\,. \tag{1.2}$$

Combining (a) and (b): To relate the cost of expert j^* and the cost of the algorithm, we combine (1.1) and (1.2) to obtain

$$\exp\Big(-\eta\sum_{i=1}^{t} c_{i,j^*}\Big) \;\leq\; \|y_{t+1}\|_1 \;\leq\; n\cdot\exp(-\epsilon A).$$

Taking the log and rearranging

$$A \;\leq\; \frac{\eta}{\epsilon}\sum_{i=1}^{t} c_{i,j^*} \;+\; \frac{\ln n}{\epsilon}.$$

As $\eta = \ln\frac{1}{1-\epsilon}$, applying Claim 3.5 concludes the proof. ■

1.1 Extensions

1.1.1 Regret bounds

The ***regret*** is defined to be the difference between the algorithm's cost and the cost of the best expert. Let $A(t)$ be the total cost of the algorithm at time step t. Then the regret at time step t is

$$\mathrm{Regret}(t) \;=\; A(t) - \min_j \sum_{i=1}^{t} c_{i,j}.$$

Remark. If the adversaries are oblivious, the regret compares the algorithm's cost to the total cost of the best fixed expert. However, if the adversary is adaptive, the cost functions are chosen based on the algorithm's behavior, and so fixing an expert may result in completely different cost functions. So, with adaptive adversaries, it's not clear if regret has any useful interpretation.

If one wishes to achieve low regret at (or slightly before) time step t, one may optimize ϵ as a function of t and obtain the following result.

Corollary 1.2. Set $\epsilon = \sqrt{\ln(n)/t}$. Then, for any $t' \leq t$ we have $\mathrm{Regret}(t') \;\leq\; 2\sqrt{t \ln n}$.

Proof.

$$\begin{aligned}
\mathrm{Regret}(t') &= A - \sum_{i=1}^{t'} c_{i,j^*} \\
&\leq \epsilon \sum_{i=1}^{t'} c_{i,j^*} + \frac{\ln n}{\epsilon} \qquad \text{(by Theorem 1.1)} \\
&\leq \epsilon t + \frac{\ln n}{\epsilon} \qquad \text{(since } c_{i,j} \in [0,1]\text{)} \\
&= 2\sqrt{t \ln n},
\end{aligned}$$

by choice of ϵ. ■

Remark. The constant factor 2 can be improved to $2^{-1/2}$, which is optimal. See [2, Section 2.2].

2 RWM using KL divergences

Now let us consider a slight variant of Algorithm 1 and a fairly different analysis. Instead of maintaining non-increasing weights $y \in \mathbb{R}^n_{>0}$, the algorithm will instead maintain normalized weights $x \in \mathbb{R}^n_{>0}$ with $\sum_j x_j = 1$. Thus, the weights always constitute a probability distribution and can be used to determine the expected cost. The modified algorithm appears in Algorithm 3.

Algorithm 2 A slight variant of the randomized weighted majority algorithm.

```
procedure RandomizedWeightedMajorityVariant($\eta$)
    Let $x_1 = (1/n, ..., 1/n)$
    for $i \leftarrow 1, 2, ...$ do
        Follow expert $j$ with probability $x_{i,j}$.
        ▷ Expected cost incurred is $\sum_{j=1}^{n} c_{i,j} x_{i,j} = c_i^{\mathsf{T}} x_i$
        ▷ Receive cost vector $c_i$
        for $j \leftarrow 1, ..., n$ do
            $y_{i+1,j} = x_{i,j} \exp(-\eta c_{i,j})$    (decrease weight according to expert $j$'s cost)
        $x_{i+1} = y_{i+1} / \|y_{i+1}\|_1$ (normalize the weights)
```

For our present purposes, Algorithm 3 is more convenient, as our new analysis will not rely on the unnormalized weights, and will instead work primarily with distributions. (Actually the analysis below can be adapted to Algorithm 1 too. In some contexts it is useful to keep track of the unnormalized weights, e.g., Theorem 1.5 above.)

Theorem 2.1. Assume that every cost vector c_i is non-negative. For any t and any distribution $z \in [0,1]^n$,

$$\sum_{i=1}^{t} (c_i^{\mathsf{T}} x_i - c_i^{\mathsf{T}} z) \;\leq\; \frac{\ln n}{\eta} + \frac{\eta}{2} \sum_{i=1}^{t} \sum_{j} x_{i,j} c_{i,j}^2. \tag{2.1}$$

Remark. This bound, which does not assume $c_{i,j} \in [0,1]$ is (slightly better than) the bound stated in [5, Theorem 1.5] for the Hedge algorithm.

2.1 Preliminaries on KL-divergence

The ***Kullback-Leibler divergence*** is a measure of how two probability distributions differ. It is also called ***KL divergence*** and ***relative entropy***.

Definition 2.2 (Discrete KL divergence). Let P and Q be discrete distributions on a countable set Ω. Suppose that $Q(x) = 0$ implies $P(x) = 0$. Then the KL-divergence is

$$D_{\mathrm{KL}}(P \parallel Q) \;=\; \sum_{x \in \Omega} P(x) \ln \frac{P(x)}{Q(x)}.$$

Note. This definition of the KL divergence uses the natural logarithm. It is said that this corresponds to using "nats" as the unit of the KL divergence.

Note. KL divergence fails to be a metric because (a) it is not symmetric, and (b) it does not satisfy the triangle inequality.

Lemma 2.3. $D_{\mathrm{KL}}(P \parallel Q) \geq 0$. Furthermore equality holds iff $P = Q$.

References. [3, Theorem 2.6.3].

Proof. Without loss of generality, $P(i) > 0$ for all $i \in \Omega$. Define $x_i = Q(i)/P(i)$ and $f(x) = -\ln x$, which is convex. By Jensen's inequality,

$$D_{\mathrm{KL}}(P \parallel Q) \;=\; -\sum_{i\in\Omega} P(i)\ln\frac{Q(i)}{P(i)} \;=\; \sum_{i\in\Omega} P(i)f(x_i) \;\geq\; f\left(\sum_{i\in\Omega} P(i)x_i\right) \;=\; f\left(\sum_{i\in\Omega} Q(i)\right) \;=\; 0.$$

Since f is strictly convex, equality holds only when there exists c such that $c = x_i = Q(i)/P(i)$ for all i. In that case $1 = \sum_i Q(i) = c\sum_i P(i) = c$, so $P = Q$. ■

Claim 2.4. Suppose that P is supported on $[n]$, and let Q be the uniform distribution on $[n]$. Then $D_{\mathrm{KL}}(P \parallel Q) = \ln n - H(P)$.

Proof.

$$D_{\mathrm{KL}}(P \parallel Q) \;=\; \sum_{i=1}^{n} p_i \ln(p_i/q_i) \;=\; \sum_{i=1}^{n} p_i \ln(n) + \sum_{i=1}^{n} p_i \ln p_i \;=\; \ln(n) - H(P).$$

■

Now we generalize the definition to non-negative vectors that are not necessarily distributions.

Definition 2.5 (Generalized KL divergence). Let $p, q \in \mathbb{R}^n_{\geq 0}$. Suppose that $q_i = 0$ implies $p_i = 0$. Then the generalized KL-divergence is

$$D_{\mathrm{KL}}(p \parallel q) \;=\; \sum_{i=1}^{n} \big(p_i \ln\frac{p_i}{q_i} \;-\; p_i \;+\; q_i\big).$$

Claim 2.6. Let $w, p \in \mathbb{R}^n_{\geq 0}$ and suppose that $\|w\|_1 = 1$. Let $\pi = p/\|p\|_1$. Then $D_{\mathrm{KL}}(w \parallel p) \geq D_{\mathrm{KL}}(w \parallel \pi)$.

Proof.

$$\begin{aligned} D_{\mathrm{KL}}(w \parallel p) - D_{\mathrm{KL}}(w \parallel \pi) \;&=\; \sum_j \left(w_j \ln\frac{w_j}{p_j} \cancel{-w_j} + p_j - w_j \ln\frac{w_j}{\pi_j} \cancel{+w_j} - \pi_j\right) \\ &=\; \sum_j \left(w_j \ln\frac{\pi_j}{p_j} + p_j - \pi_j\right) \\ &=\; \ln\frac{1}{\|p\|_1} + \|p\|_1 - 1 \end{aligned}$$

This is non-negative by Claim 3.4. ■

2.2 Proof of Theorem 2.1

Proof. The main idea with this proof is to analyze, at each iteration i, the difference between the algorithm's cost and the cost of some other strategy (denoted by the distribution z). For any distribution z,

$$c_i^{\mathsf{T}} x_i - c_i^{\mathsf{T}} z \;=\; c_i^{\mathsf{T}}(x_i - z)$$

For notational convenience, let $\ln x_i$ and $\ln y_{i+1}$ be the vectors obtained by taking entry-wise logarithm. Since $y_{i+1,j} = x_{i,j}\exp(-\eta c_{i,j})$, we have $c_{i,j} = (\ln x_{i,j} - \ln y_{i+1,j})/\eta$, so

$$\begin{aligned}
&= \frac{1}{\eta}(\ln x_i - \ln y_{i+1})^{\mathsf{T}}(x_i - z) \\
&= \frac{1}{\eta}\sum_{j=1}^{n}\Big(x_{i,j}\ln\frac{x_{i,j}}{y_{i+1,j}} - z_j \ln\frac{x_{i,j}}{y_{i+1,j}}\Big) \\
&= \frac{1}{\eta}\sum_{j=1}^{n}\Big(x_{i,j}\ln\frac{x_{i,j}}{y_{i+1,j}} - z_j \ln\frac{z_j}{y_{i+1,j}} + z_j\ln\frac{z_j}{x_{i,j}}\Big) \\
&= \frac{1}{\eta}\big(D_{\mathrm{KL}}(x_i \parallel y_{i+1}) - D_{\mathrm{KL}}(z \parallel y_{i+1}) + D_{\mathrm{KL}}(z \parallel x_i)\big) \\
&\leq \frac{1}{\eta}\big(D_{\mathrm{KL}}(x_i \parallel y_{i+1}) - D_{\mathrm{KL}}(z \parallel x_{i+1}) + D_{\mathrm{KL}}(z \parallel x_i)\big) \qquad \text{(by Claim 2.6)}.
\end{aligned}$$

(Note that these are generalized KL divergences as y_{i+1} is not necessarily a distribution.) Summing this over i, the last two terms telescope:

$$\sum_{i=1}^{t}(c_i^{\mathsf{T}}x_i - c_i^{\mathsf{T}}z) \;\leq\; \frac{1}{\eta}\Big(D_{\mathrm{KL}}(z \parallel x_1) + \sum_{i=1}^{t} D_{\mathrm{KL}}(x_i \parallel y_{i+1})\Big) \tag{2.2}$$

The summands on the right-hand side are bounded by direct expansion:

$$\begin{aligned}
D_{\mathrm{KL}}(x_i \parallel y_{i+1}) &= \sum_j \Big(x_{i,j}\ln\frac{x_{i,j}}{y_{i+1,j}} - x_{i,j} + y_{i+1,j}\Big) \\
&= \sum_j x_{i,j}\cdot\big(\eta c_{i,j} - 1 + \exp(-\eta c_{i,j})\big) \\
&\leq \sum_j x_{i,j}\cdot \eta^2 c_{i,j}^2/2, \qquad (2.3)
\end{aligned}$$

by Claim 3.3. Combining this with (2.2) and applying Claim 2.4 yields

$$\sum_{i=1}^{t}(c_i^{\mathsf{T}}x_i - c_i^{\mathsf{T}}z) \;\leq\; \frac{\ln n}{\eta} + \frac{\eta}{2}\sum_{i=1}^{t}\sum_j x_{i,j}c_{i,j}^2.$$

■

Corollary 2.7. Assume that the costs satisfy $c_{i,j} \in [0,1]$ for all i,j. Let $\epsilon = \eta - \eta^2/2$ and assume that $\epsilon \leq \sqrt{2} - 1 \approx 0.414$. Consider any time step t. Let A be the total expected cost of the algorithm. Let j^* be the expert with minimum total cost. Then

$$A \;\leq\; (1+\epsilon)\sum_{i=1}^{t} c_{i,j^*} + \frac{\ln n}{\epsilon}. \tag{2.4}$$

Remark. The bound (2.4) is the same as Corollary 2.7, with a slightly different relationship between ϵ and η.

Proof. Apply Theorem 2.1 with $z = e_{j^*}$, obtaining

$$\sum_{i=1}^{t} c_i^\mathsf{T} x_i \;\leq\; \sum_{i=1}^{t} c_{i,j^*} + \frac{\eta}{2} \sum_{i=1}^{t} \sum_{j} x_{i,j} c_{i,j}^2 + \frac{\ln n}{\eta}.$$

Since $c_i \in [0,1]^n$ we have $c_{i,j}^2 \leq c_{i,j}$. So, rearranging, we obtain

$$\sum_{i=1}^{t} c_i^\mathsf{T} x_i \;\leq\; \frac{1}{1-\eta/2} \sum_{i=1}^{t} c_{i,j^*} + \frac{\ln n}{(1-\eta/2)\eta}.$$

Recall that $(1-\eta/2)\eta = \epsilon$. Since $\epsilon \leq \sqrt{2}-1$, we have $\frac{1}{1-\eta/2} \leq 1+\epsilon$. This proves (2.4). ■

Corollary 2.8. Assume that the costs satisfy $c_{i,j} \in [0,1]$ for all i, j. Let $\epsilon = \sqrt{\ln(n)/t}$. Consider any time step t. Let A be the total expected cost of the algorithm. Let j^* be the expert with minimum total cost. Then

$$\operatorname{Regret}(t) \;=\; A - \sum_{i=1}^{t} c_{i,j^*} \;\leq\; 2\sqrt{t \ln n}. \qquad (2.5)$$

3 Basic Facts

Claim 3.1. $1 + x \leq e^x$ for all $x \in \mathbb{R}$.

Claim 3.2. For any $\alpha > 0$,

$$\alpha^x \;\leq\; 1 + (\alpha - 1)x \qquad \forall x \in [0,1].$$

Claim 3.3. $e^{-x} \leq 1 - x + \frac{x^2}{2}$ for $x \geq 0$.

Proof. Observe that both sides equal 0 when $x = 0$. The derivative of $1-x+\frac{x^2}{2}-e^{-x}$ is $e^{-x}-1+x$. This is non-negative by Claim 3.1. Integrating proves the result. ■

Claim 3.4.

$$\log(1+x) \;\leq\; x \qquad \forall x > -1.$$

Claim 3.5.

$$\log\frac{1}{1-x} \;\leq\; x + x^2 \qquad \forall x \in [0, 1/2]$$

References

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[2] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning and Games*. Cambridge University Press, 2006.

[3] Thomas Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991.

[4] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[5] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3–4), 2015.

[6] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.