EECS598: Prediction and Learning Fall 2013

Lecture 3: The Exponential Weights Algorithm

Prof. Jacob Abernethy *Scribe: Cat Saint Croix*

Announcements

- Our GSI is Sindhu Kutty, skutty@umich.edu
- The homework has been posted and is **due 9/25**.
- EECS598 now has a CTools site; submit homework there.

3.1 Reviewing the Weighted Majority Algorithm

In Lecture 2, we introduced the Weighted Majority Algorithm, which allows us to drop the very strong "realizability" assumption[1] used in the Halving algorithm. In doing so, we proved the following theorem:

Theorem 3.1. *For any expert i and $\epsilon \le \frac{1}{2}$*

$$\textsc{Mistakes}_T(WMA) \;\le\; \frac{2\log(n)}{\epsilon} + 2(1+\epsilon)\,\textsc{Mistakes}_T(expert_i)$$

Note that this restriction on ϵ arises because one of the inequalities we use in proving this is restricted to $x \in [0, \frac{1}{2}]$.

A note on tuning parameters like ϵ: What's the best choice for an ϵ? Minimize w.r.t ϵ by choosing ϵ as a function of the inputs. For example, given inputs A and B s.t. performance $= \frac{A}{\epsilon} + B\epsilon$, the optimal value for ϵ, $\epsilon^* = \sqrt{\frac{A}{B}}$. With this ϵ, performance = $2\sqrt{AB}$. The notion is that the terms should grow at the same rate (so, one can approximate the optimization by setting the terms, $\frac{A}{\epsilon}$ and $B\epsilon$, equal to one another). Tricks to this sort of (psuedo-)optimization show up on the homework.

Now, applying this to the inequality above, with $A = 2\log(n)$ and $B = 2\textsc{Mistakes}_T(expert_i)$, we get

$$\textsc{Mistakes}_T(WMA) \;\le\; 2\textsc{Mistakes}_T(expert_i) + 4\sqrt{\log(n)\textsc{Mistakes}_T(expert_i)} \qquad (3.1)$$

The two in equation (3.2) indicates a higher cost than one would think necessary in a situation like this. The algorithm we discuss in this lecture lets us get rid of this high cost.

In the process of proving **Theorem 3.1**, we used some approximations, which will be useful again in the case of the Exponential Weight Algorithm. These are:

[1] That there exists a perfect expert.

(1) Lower bound on the logarithm function:

$$log(1+x) \leq x \qquad \forall x \tag{3.2}$$

(2) Upper bound

$$e^{\alpha x} \leq 1 + (e^{\alpha} - 1)x \qquad x \in [0,1] \tag{3.3}$$

(3)

$$-\,log(1-x) \leq x + x^2 \qquad x \in [0, \frac{1}{2}] \tag{3.4}$$

3.2 Introduction to Loss Functions

How do we measure the quality of a prediction given an outcome? Use a *loss function*! For a loss function $\ell(\hat{y}, y)$, ℓ describes the "cost" of guessing $\hat{y}$ when the outcome is y.

Here, we will assume convexity for the time being. Why?

- Convex functions are well-behaved
- Demonstrates an interesting property of prediction – intuitively, the average of two predictions shouldn't do worse than both. So, it's a natural choice.

Let ℓ be a convex loss function,

$$\ell : [0,1] \times \{0,1\} \to [0,1] \tag{3.5}$$

Some examples of loss functions:

(1) Squared Loss

$$\ell(\hat{y}, y) = (\hat{y} - y)^2 \tag{3.6}$$

(2) Absolute Loss

$$\ell(\hat{y}, y) = |\hat{y} - y| \tag{3.7}$$

(3) Log Loss

$$\ell(\hat{y}, y) = \begin{cases} \log(\hat{y}), & \text{if } y = 1 \\ \log(1-\hat{y}), & \text{if } y = 0 \end{cases} \tag{3.8}$$

Note: log loss violates the $[0,1]$ bound used in our definition of a loss function above, so it won't be used in the following discussion.

Note that (1) and (3) are proper loss functions: the true probability minimizes expected loss.

3.3 The Exponential Weight Algorithm

In plain(-ish) English: Given a set of experts and their predictions, the EXPONENTIAL WEIGHT ALGORITHM[2] begins with equal weights for each. On each round, it makes a prediction based on the predictions of these experts (according to their weights). After learning the outcome of that round, the algorithm reduces the weight of incorrect experts by multiplying their previous weights by the exponential defined in the algorithm below.

Algorithm 1: EXPONENTIAL WEIGHT ALGORITHM

```
Input: $N$ experts $i$ each predicting outcomes $f_i^t$ for round $t$, a parameter $\eta$
$w_i^1 \longleftarrow 1$ for $i = 1, ..., N$      (set initial weight of each expert to 1)
for rounds $t = 1, 2, ...$ do
    $\hat{y}_t \longleftarrow \frac{\sum_{i=1}^N w_i^t f_i^t}{\sum_{i=1}^N w_i^t}$      (compute prediction)
    Outcome $y_t$ is revealed
    $w_i^{t+1} \longleftarrow w_i^t e^{-\eta \ell(f_i^t, y_t)}$      (update the weight assigned to each expert)
end
```

Theorem 3.2. *Let L_{MA}^t (respectively, L_i^t) be the accumulated losses (up to round t) for the master algorithm (MA) (respectively, expert i). That is,*

$$L_{MA}^t := \sum_{s=1}^t \ell(\hat{y}_s, y_s), \qquad L_i^t := \sum_{s=1}^t \ell(f_i^s, y_s)$$

Then, the following bound holds on the loss of MA over T rounds

$$L_{MA}^T \leq \frac{\eta L_i^T + \log N}{1 - e^{-\eta}}$$

We begin by proving a lemma:

Lemma 3.3. *For any s and r.v. X taking values in* $[0, 1]$*,*

$$\log \mathbb{E}[e^{sX}] \leq (e^s - 1)\mathbb{E}X \tag{3.9}$$

Proof of Lemma: By definition,

$$\mathbb{E}(e^{sX}) = \sum_x \Pr(x) e^{sX}$$

by inequality 3.3,

$$\sum_x \Pr(x) e^{sX} \leq \sum_x \Pr(x)(1 + (e^s - 1)x)$$

Further, since $\sum_x \Pr(x) = 1$ and $\sum_x x \Pr(x) = \mathbb{E}[x]$, we have

$$\mathbb{E}e^{sX} \leq \mathbb{E}X(e^s - 1) + 1$$

[2] a.k.a. the EXPONENTIALLY WEIGHTED AVERAGE FORECASTER. See Lugosi & Gabòr, 2006, *Prediction, Learning, and Games*, p.14

Since log is monotonic, it follows that

$$\log(\mathbb{E}e^{sX}) \leq \log(1 + \mathbb{E}X(e^s - 1))$$

Using inequality 3.2, we have

$$\log(\mathbb{E}e^{sX}) \leq \mathbb{E}X(e^s - 1)$$

□

We are now ready to prove the theorem.

Proof of Theorem 3.2:
We will use a potential function:

$$\Phi_t = -\log W_t, \tag{3.10}$$

where

$$W_t = \sum_{i=1}^{N} w_i^t \tag{3.11}$$

Look at the increase in potential in a round:

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= -\log W_{t+1} + \log W_t \\
&= -\log \frac{W_{t+1}}{W_t} \\
&= -\log \frac{\sum_{i=1}^{N} w_i^t e^{-\eta \ell_i^t}}{\sum_{i=1}^{N} w_i^t} \qquad (3.12)
\end{aligned}$$

where we define $\ell_i^t := \ell(f_i^t, y_t)$ and equality 3.12 follows from the fact that $w_i^{t+1} = w_i^t e^{-\eta \ell(f_i^t, y_t)}$ (see algorithm definition).

Now, define r.v. X taking value ℓ_i^t with probability $\frac{w_i^t}{\sum_i w_i^t}$. Then applying Lemma 3.3

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= -\log \mathbb{E}e^{-\eta X} \\
&\geq -(e^{-\eta} - 1)\mathbb{E}X \\
&= (1 - e^{-\eta}) \frac{\sum_i w_i^t \ell_i^t}{\sum_i w_i^t} \\
&= (1 - e^{-\eta}) \frac{\sum_i w_i^t \ell(f_i^t, y_t)}{\sum_i w_i^t} \\
&\geq (1 - e^{-\eta}) \ell\left(\frac{\sum_i w_i^t f_i^t}{\sum_i w_i^t}, y_t\right) \qquad (3.13) \\
&= (1 - e^{-\eta}) \ell(\hat{y}_t, y_t) \qquad (3.14)
\end{aligned}$$

where 3.13 follows from the fact that ℓ is a **convex loss function**, meaning that the loss of the

MA's average must be *less* than the average of their losses. Now,

$$\begin{aligned}
(1-e^{-\eta})L_{MA}^{T} &= (1-e^{-\eta})\sum_{t=1}^{T}\ell(\hat{y}_t, y_t) \\
&\leq \sum_{t=1}^{T}\Phi_{t+1}-\Phi_t \quad \text{(from 3.14)} \\
&= \Phi_{T+1}-\Phi_1 \quad \text{(telescoping sum)} \\
&= -\log\sum_i w_i^{T+1}+\log N \\
&\leq -\log w_i^{T+1}+\log N \quad \text{(for any expert } i\text{)} \\
&= -\log[e^{-\eta\sum_{t=1}^{T}\ell_i^t}]+\log N \\
&= \eta L_i^T+\log N
\end{aligned}$$

Now, dividing both sides by $(1-e^{-\eta})$, we get

$$L_{MA}^{T} \leq \frac{\eta L_i^T+\log N}{1-e^{-\eta}}$$

□

Corollary 3.4. *For an optimally-tuned η,*

$$L_{MA}^{T} \quad \leq \quad \underbrace{L_i^T}_{\text{loss of the best expert}} \quad + \quad \overbrace{\log N}^{\text{Halving algorithm cost}} \quad + \underbrace{\sqrt{2L_i^T\log N}}_{\text{additional cost}} \tag{3.15}$$

See homework regarding tuning parameters...

3.4 Proof Techniques

The above proof relies on a particular proof technique involving **potential functions**. Here, we had $\Delta\ell(alg.) \leq \Delta\Phi$, where Φ was a path-independent sufficient statistic on the cost of the algorithm to T. This tells us roughly where the loss of the algorithm will be, depending only on the cumulative loss of the experts. This allows us to bound the cost based on the potential function.

Additionally, the potential function is softmin, $-\frac{1}{\eta}\log(\sum_i e^{-\eta x_i})$, which is close to the loss of the best.

3.5 Next time...

So far, the algorithms in play have used "experts". Moving forward, we'll talk in terms of actions. Suppose, instead of experts, we have plans of action. Now, if, instead of continuing the proof after the use of the convexity assumption, we use a random strategy over actions, we have the expected loss of the choice. We can use this fact to talk in terms of game theory rather than prediction!