CS 7545: Machine Learning Theory **Fall 2019**

Lecture 7: Exponential Weights Algorithm

Lecturer: Jacob Abernethy *Scribes: Ankit Arora and Hezi Zhang*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

7.1 Previously: Weighted Majority

In previous lecture we introduced online learning problems and analysed the bounds of Weighted Majority Algorithm (WMA)

Algorithm 1: Weighted Majority Algorithm (WMA)

Parameter $\epsilon \in (0, 1)$;
There are N experts making predictions;
We maintain weights w_i, $\forall$ expert $i = 1, 2, \ldots, N$;
Initialize weights $w_i^1 = 1$ for i $\in \{1, ..., N\}$;
for $t = 1$ *to* T **do**
  Expert i predicts $x_i^t \in [0, 1]$ for $i \in 1, ..., N$;
  Predicts $\hat{y}^t = round\left(\frac{\sum_{i=1}^N w_i^t x_i^t}{\sum_{j=1}^N w_j^t}\right)$;
  Nature reveals $y^t \in [0, 1]$; s.t loss function $l(\hat{y}^t, y^t)$ ($l : [0, 1] \times \{0, 1\} \to \mathbb{R}$);
  Update $w_i^{t+1} = w_i^t (1 - \epsilon)^{\mathbb{I}[x_i^t \neq y_t]}$
end

For any expert i and $\epsilon \in (0, \frac{1}{2})$, WMA guarantees,

$$\#M(WMA) \leq \frac{2 \log_e N}{\epsilon} + 2(1 + \epsilon) M_T(i)$$

If we set $\epsilon = \sqrt{\frac{\log N}{\#M(i^*)}}$, where i^* is the best expert with minimum number of mistakes, then

$$\#M(WMA) \leq 2\#M(i^*) + 4\sqrt{(\log N)\#M(i^*)}$$

In this lecture, we will introduce Exponential Weights Algorithm a generalized variant of WMA, and will show a tighter bound similar to that of WMA.

7.2 Online Learning Frameworks

Before we go through Exponential weighted average Algorithm we will describe two key settings for online learning framework.

7.2.1 Setting 1: Expert Advice or Continuous Prediction

For a convex loss function $l(.)$ ($l : [0, 1]$ x $\{0, 1\} \to \mathbb{R}$), a pool of N experts, and an algorithm $\mathcal{A}$, an online learning framework with Expert Advice is following.

Algorithm 2: Expert Advice Framework

for t = *1 to T* **do**
- Expert i predicts $x_i^t \in [0,1]$;
- Algorithm $\mathcal{A}$ predicts $\hat{y}^t \in [0,1]$;
- Nature reveals $y^t \in [0,1]$;
- Loss of algorithm at t is $l(\hat{y}^t, y^t)$;

end

Examples of loss functions:

- absolute loss: $l(\hat{y}, y) = |\hat{y}, y|$
- square loss: $l(\hat{y}, y) = (\hat{y} - y)^2$
- log loss: $l(\hat{y}, y) = -y \log \hat{y} - (1-y)\log(1-\hat{y})$

Definition 7.1 ***(Algorithm loss)*** *The* ***loss*** $L_T(Alg)$ *at time* T *is the sum of the loss values* $l(\hat{y}^t, y^t)$ *from* t *= 1 to* T*.*

$$L_T(Alg) := \sum_{t=1}^{T} l(\hat{y}^t, y^t)$$

Definition 7.2 ***(Expert i loss)*** *The* ***loss*** $L_T(i)$ *of an expert* i *at time* T *is the sum of the loss values* $l(x_i^t, y^t)$ *from* t *= 1 to* T*.*

$$L_T(i) := \sum_{t=1}^{T} l(x_i^t, y^t)$$

7.2.2 Setting 2: Hedge/Action Framework

The Hedge framework, in which at each timestep, instead of N experts, there are N possible actions from which to choose, and a loss associated with each action at that time step. For N actions, and an algorithm A, the Hedge framework is the following framework.

Algorithm 3: Hedge/Action Framework

for t = *1 to T* **do**
- There are N actions;
- Algorithm must (randomly) select an action i_t on day t;
- Equivalently: Algorithm selects $p^t \in \Delta_N$;
- Then nature chooses losses $l^t = [l_1^t, l_2^t, \ldots, l_N^t] \in [0,1]^N$, where l_i^t is the cost of choosing i at t;
- Expected cost to the algorithm is $p^t \cdot l^t = E_{i \sim p^t}[l_i^t]$;

end

Definition 7.3 ***(Algorithm loss in Hedge framework)*** *The* ***loss*** $L_T(Alg)$ *at time* T *is the sum of the cost* $(p^s \cdot l^s)$ *from* s *= 1 to* T*.*

$$L_T(Alg) = \sum_{s=1}^{T} p^s \cdot l^s$$

Definition 7.4 ***(Action loss)*** *The* ***loss*** $L_T(i)$ *of an action* i *at time* T *is the sum of the cost* (l_i^s) *from s = 1 to* T*.*

$$L_T(i) = \sum_{s=1}^{T} l_i^s$$

Definition 7.5 ***(Regret)*** *The regret$R_T(Alg)$ at time T is the difference between Algorithm loss at time T and the loss of the best action at time T.*

$$\text{Regret}_T(Alg) = L_T(Alg) - \min_{i \in [N]} L_T[i]$$

7.3 Exponential Weights Algorithm

Algorithm 4: Exponential Weight Algorithm

Data: $\eta \in (0,1)$
Initialize weights $w_i^1 = 1$ for i $\in \{1, ..., N\}$;
for $t = 1$ *to* T **do**
  if *Expert Framework* **then**
    Expert i predicts $x_i^t \in [0,1]$ for $i \in 1, ..., N$;
    Predicts $\hat{y}^t = \frac{\sum_{i=1}^N w_i^t x_i^t}{\sum_{j=1}^N w_j^t}$;
    Nature reveals $y^t \in [0,1]$;
    $l_i^t = l\left(x_i^t, y^t\right)$;
  else if *Hedge Framework* **then**
    set $p_i^t = \frac{w_i^t}{\sum_{j=1}^N w_j^t}$;
    Nature reveals $l_i^t \in [0,1]$;
    Algorithm pays cost $p^t \cdot l^t = \mathbb{E}_{i \sim p^t}[l_i^t]$;
  Update weights according to $w_i^{t+1} = w_i^t exp(-\eta l_i^t)$;
end

We will now show that the EWA guarantees a regret(in the experts framework) that is similar to mistake bound of the WMA.

Theorem 7.6 *Assume that the loss function L is convex and takes values in [0,1]. Then, For any $\eta > 0$ and any sequence of inputs, $\mathbf{EWA}(\eta)$ guarantees that*

$$L_T(\mathbf{EWA}(\eta)) \leq \frac{log(N) + \eta L_T(i)}{1 - e^{-\eta}}$$

Corollary 7.7 *For excellent choice of $\eta > 0$ for $\forall$ i*

$$L_T(\mathbf{EWA}(\eta)) - L_T(i^*) \leq \log(N) + \sqrt{2L_T(i^*)\log(N)}$$

where i^* is best expert such that $i^* = argmin_i L_T(i)$

Lemma 7.8 *For any value of $x \in [0,1]$*

$$e^{sx} \leq 1 + (e^s - 1)x$$

(This Lemma was proved in the last lecture.)

Lemma 7.9 *For any r.v $X \in [0,1]$ and any $s \in \mathbb{R}$,*

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

Proof: From Lemma 7.8,

$$e^{sX} \leq 1 + (e^s - 1)x; \forall x \in [0,1]$$

Taking expectation on both sides,

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

Taking log on both sides,

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

As $log(1+x) \leq x, thus$

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

■

Proof: Similar to the proof of the mistake bound for WMA, we will use a potential function Φ, where

$$\Phi_t = -\log(\sum_{i=1} N W_i^t)$$

.

Lower bound for difference $\Phi_{t+1} - \Phi_t$ is,

$$\Phi_{t+1} - \Phi_t = -\log\left(\frac{\sum_{i=1}^N W_i^t}{\sum_{j=1}^N W_j^t}\right) = -\log\left(\frac{\sum_{i=1}^N W_i^t \exp(-\eta l(x_i^t, y^t))}{\sum_{j=1}^N W_j^t}\right)$$

For each t, let X_t be a random variable which takes the value $l(x_i^t, y^t)$ with probability $\frac{W_i^t}{\sum_{k=1}^N W_k^t}$, Thus

$$\Phi_{t+1} - \Phi_t = -\log\left(\frac{\sum_{i=1}^N W_i^t \exp(-\eta l(x_i^t, y^t))}{\sum_{j=1}^N W_j^t}\right) = -\log(\mathbb{E}[e^{-\eta X_t}])$$

From Lemma 7.9

$$\Phi_{t+1} - \Phi_t \geq (1 - e^{-\eta})\mathbb{E}[X_t] = (1 - e^{-\eta}) \sum_{i=1}^N \frac{W_i^t}{\sum_{j=1}^N W_j^t} l(x_t^t, y^t)$$

Apply Jensen's inequality

$$\begin{aligned}\Phi_{t+1} - \Phi_t &\geq (1 - e^{-\eta}) l\left(\sum_{i=1}^N \frac{W_i^t x_t^t}{\sum_{j=1}^N W_j^t}, y^t\right) \\ &\geq (1 - e^{-\eta}) l(\hat{y}, y_t)\end{aligned}$$

Note that

- $\Phi_1 = -\log(N)$
- $\Phi_{T+1} \leq -log(\sum_{i=1}^N exp(-\eta L_T(i)) \leq \eta L_T(i)$, for any $i \in 1, ..., N$

Thus,

$$\begin{aligned}\log(N) + \eta L_T(i) \geq \Phi_{T+1} - \Phi_1 = \sum_{i=1}^T (\Phi_{t+1} - \Phi_t) &\geq (1 - e^{-\eta})(\sum_{t=1}^T l(\hat{y}^t, y^t)) \\ &\geq (1 - e^{-\eta}) L_T(EWA(\eta))\end{aligned}$$

Hence,

$$L_T(\mathbf{EWA}(\eta)) \leq \frac{log(N) + \eta L_T(i)}{1 - e^{-\eta}}$$

■