EE6550 Machine Learning

Lecture Seven – On-Line Learning

Chung-Chin Lu

Department of Electrical Engineering

National Tsing Hua University

April 17, 2017

Motivations

- PAC learning:
  - Distribution is fixed but unknown over time (training and testing).
  - IID assumption for samples.
- On-line learning:
  - No distributional assumption.
  - Worst-case analysis (adversarial).
  - Mixed training and testing.
  - Performance measure: mistake model, regret.

The Contents of This Lecture

- Prediction with expert advice.
- Linear classification.
- On-line to batch conversion.

General On-Line Setting

- For $t = 1$ to T do
  - receive instance $\omega_t \in \mathscr{I}$;
  - predict $\hat{y}_t \in \mathscr{Y}'$;
  - receive label $c(\omega_t) \in \mathscr{Y}$ of ω_t;
  - incur loss $L(\hat{y}, c(\omega_t))$, where $L : \mathscr{Y}' \times \mathscr{Y} \to \mathbb{R}^+$ is the loss function.
- Classification: $\mathscr{Y}' = \mathscr{Y} = \{0, 1\}$ and $L(y', y) = 1_{y' \neq y}$.
- Regression: $\mathscr{Y}' = \mathscr{Y} = \mathbb{R}$ and $L(y', y) = (y' - y)^2$.
- Objective: minimizing the total loss $\sum_{t=1}^{T} L(\hat{y}_t, c(\omega_t))$.

Prediction with Expert Advice

- For $t = 1$ to T do
  - receive instance $\omega_t \in \mathscr{I}$ and advice $\hat{y}_{t,i} \in \mathscr{Y}$, $i \in [1, N]$;

  - predict $\hat{y}_t \in \mathscr{Y}'$;
  - receive label $c(\omega_t) \in \mathscr{Y}$ of ω_t;
  - incur loss $L(\hat{y}, c(\omega_t))$.
- Objective: minimizing the (external) regret R_T, i.e., difference of total loss incurred and that of best expert,

$$R_T \triangleq \sum_{t=1}^{T} L(\hat{y}_t, c(\omega_t)) - \min_{i \in [1,N]} \sum_{t=1}^{T} L(\hat{y}_{t,i}, c(\omega_t)).$$

On-line Learning a Realizable Concept

- $\mathcal{H}$: the hypothesis set used by an on-line learning algorithm $\mathbb{A}$.
- C : a concept class to learn by $\mathbb{A}$.
- Assumption : $C \subseteq \mathcal{H}$.
- Question : How many mistakes before we learn a particular concept $c \in C$?
  - Since we are in the realizable case, i.e., we have assumed $C \subseteq \mathcal{H}$, after some number of rounds T, we will learn the concept and no longer make mistakes in subsequent rounds.

Mistake Bound Model for Realizable On-line Learning

- C : a concept class to learn which is assumed to be a subset of the hypothesis set $\mathcal{H}$.
- $S = (\omega_1, \ldots, \omega_T)$: a sample of T items with labels $(c(\omega_1), \ldots, c(\omega_T))$ from a fixed but unknown concept c.
- $\mathbb{A}$: an on-line learning algorithm which will output the target concept c from the hypothesis set $\mathcal{H}$ after T rounds based on the sample S.
- mistakes$(\mathbb{A}, c; S) = \{\omega_t, t \in [1, T] \mid \hat{y}_t \neq c(\omega_t)\}$: the set of all items in S for which the on-line algorithm $\mathbb{A}$ makes mistake before the target concept c is learned.

- $M_{\mathbb{A}}(c)$: the maximum number of mistakes the on-line learning algorithm $\mathbb{A}$ makes to learn a particular concept c,

$$M_{\mathbb{A}}(c) \triangleq \max_{S} |\text{mistakes}(\mathbb{A}, c; S)|.$$

- $M_{\mathbb{A}}(C)$: the maximum number of mistakes the on-line learning algorithm $\mathbb{A}$ makes to learn an arbitrary concept in the concept class C,

$$M_{\mathbb{A}}(C) \triangleq \max_{C} M_{\mathbb{A}}(c).$$

- A mistake bound of an on-line learning algorithm $\mathbb{A}$ for a concept class C is a bound for $M_{\mathbb{A}}(C)$.

The Halving Algorithm

HALVING($\mathcal{H}$) ▷ $\mathcal{H}$ is the hypothesis set of the learning algorithm

1. $\mathcal{H}_1 \leftarrow \mathcal{H}$
2. **for** $t \leftarrow 1$ **to** T **do**
3. RECEIVE(ω_t)
4. $\hat{y}_t \leftarrow$ MAJORITYVOTE($\mathcal{H}_t, \omega_t$)
5. RECEIVE($c(\omega_t)$)
6. **if** $(\hat{y}_t \neq c(\omega_t))$ **then**
7. $\mathcal{H}_{t+1} \leftarrow \{h \in \mathcal{H}_t \mid h(\omega_i) = c(\omega_i)\}$
8. **else**
9. $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t$
10. **return** $\mathcal{H}_{T+1}$

Finite Hypothesis Set – Realizable Concepts

Theorem 7.1: Let

- $\mathbb{A} =$ Halving : the Halving algorithm is the on-line learning algorithm;
- $\mathcal{H}$: the hypothesis set of the on-line learning algorithm which is finite;
- $C = \mathcal{H}$.

Then

$$M_{\text{Halving}}(\mathcal{H}) \leq \log_2 |\mathcal{H}|.$$

Proof. It is clear from the Halving algorithm. □

Finite Hypothesis Set – Realizable Concepts

Theorem 7.2: Let

- $\mathcal{H}$: the hypothesis set of the on-line learning algorithm which is finite;
- $C = \mathcal{H}$;
- $\text{opt}(\mathcal{H})$: the optimal mistake bound for $\mathcal{H}$.

Then,

$$\text{VCdim}(\mathcal{H}) \leq \text{opt}(\mathcal{H}) \leq M_{\text{Halving}}(\mathcal{H}) \leq \log_2 |\mathcal{H}|.$$

Proof. To prove the first inequality, we let $d = \text{VCdim}(\mathcal{H})$.

- There exists a fully shattered set of d items by the hypothesis set $\mathcal{H}$.
- With such a fully shattered set of d items, we can form a

complete binary tree of expert advice with height d.

- Since in the worst case, there is no consensus among experts at any node of the complete binary tree of advice, we may choose labels at each round of learning so that d mistakes are made in d rounds.

Note that this adversarial argument is valid since the on-line setting makes no statistical assumptions about the data. $\square$

On-line Learning a Non-Realizable Concept

- N : the number of experts used by an on-line learning algorithm $\mathbb{A}$.
  - These N experts form the hypothesis set $\mathcal{H}$ of the on-line algorithm $\mathbb{A}$.
- C : a concept class to learn by $\mathbb{A}$.
  - In general, $C \nsubseteq \mathcal{H}$.
- Questions : How many mistakes m_T after T rounds of on-line learning ? What is the regret R_T ?
  - Since we are in the non-realizable case, we will continue to make mistakes in subsequent rounds.

The Weighted Majority Algorithm

WEIGHTED-MAJORITY(N)

1. **for** $i \leftarrow 1$ **to** N **do**
2. $\quad w_{1,i} \leftarrow 1$
3. **for** $t \leftarrow 1$ **to** T **do**
4. $\quad$ RECEIVE(ω_t)
5. $\quad$ **if** $\sum_{i:\hat{y}_{t,i}=1} w_{t,i} \geq \sum_{i:\hat{y}_{t,i}=0} w_{t,i}$ **then**
6. $\quad\quad \hat{y}_t \leftarrow 1$
7. $\quad$ **else**
8. $\quad\quad \hat{y}_t \leftarrow 0$
9. $\quad$ RECEIVE($c(\omega_t)$)
10. $\quad$ **if** $(\hat{y}_t \neq c(\omega_t))$ **then**

11. **for** $i \leftarrow 1$ **to** N **do**

12. **if** $(\hat{y}_{t,i} \neq c(\omega_t))$ **then**

13. $w_{t+1,i} \leftarrow \beta w_{t,i}$

14. **else**

15. $w_{t+1,i} \leftarrow w_{t,i}$

16. **else**

17. **for** $i \leftarrow 1$ **to** N **do**

18. $w_{t+1,i} \leftarrow w_{t,i}$

19. **return** $\mathbf{w}_{T+1}$

Remarks

- The weighted majority (WM) algorithm weights the importance of experts as a function of their mistake rate.
  - The WM algorithm begins with uniform weights over all N experts.
- At each round, the WM algorithm generates predictions using a weighted majority vote.
- After receiving the true label, the algorithm then reduces the weight of each incorrect expert by a factor of $\beta \in [0, 1)$.
  - When $\beta = 0$, the weighted majority algorithm becomes to the halving algorithm.
- m_T : the number of mistakes made by the WM algorithm after T rounds of on-line learning.

- m_T^* : the number of mistakes made by the best expert in hindsight, i.e., after T rounds of on-line learning.
- The next theorem gives a bound for m_T as a function of m_T^*.

Weighted Majority - Bound for Non-Realizable Concepts

Theorem 7.3: Let

- N : the number of experts;
- $\beta \in (0,1)$: the weight reduction factor.

Then

$$m_T \leq \frac{\ln N + m_T^* \ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}}.$$

Proof.

- $W_t \triangleq \sum_{i=1}^{N} w_{t,i}, t \in [1,T]$: the potential function.
  - $W_1 = N$.
- A mistake is made at the t-th round (if and) only if

$$\sum_{i:\hat{y}_{t,i} \neq c(\omega_t)} w_{t,i} \geq \sum_{i:\hat{y}_{t,i} = c(\omega_t)} w_{t,i}.$$

Since $W_t = \sum_{i:\hat{y}_{t,i} \neq c(\omega_t)} w_{t,i} + \sum_{i:\hat{y}_{t,i}=c(\omega_t)} w_{t,i}$, we have

$$\sum_{i:\hat{y}_{t,i} \neq c(\omega_t)} w_{t,i} = \frac{W_t}{2} + \alpha \text{ and } \sum_{i:\hat{y}_{t,i}=c(\omega_t)} w_{t,i} = \frac{W_t}{2} - \alpha$$

for some $\alpha \geq 0$. Thus we have

$$\begin{aligned} W_{t+1} &= \sum_{i:\hat{y}_{t,i} \neq c(\omega_t)} \beta w_{t,i} + \sum_{i:\hat{y}_{t,i}=c(\omega_t)} w_{t,i} \\ &= \beta \left(\frac{W_t}{2} + \alpha \right) + \frac{W_t}{2} - \alpha \\ &= \left(\frac{1+\beta}{2} \right) W_t - (1-\beta)\alpha \\ &\leq \left(\frac{1+\beta}{2} \right) W_t. \end{aligned}$$

- Since there are m_T mistakes after T rounds, we have

$$W_{T+1} \leq \left(\frac{1+\beta}{2}\right)^{m_T} W_1.$$

- $m_{T,i}, i \in [1, N]$: the number of mistakes made by expert i after T rounds.
- $W_{T+1} = \sum_{j=1}^{N} w_{T+1,j} \geq w_{T+1,i} = \beta^{m_{T,i}}$ for all $i \in [1, N]$.

Now we have

$$\beta^{m_T^*} \leq W_{T+1} \leq \left(\frac{1+\beta}{2}\right)^{m_T} N$$

and then

$$m_T \leq \frac{\ln N + m_T^* \ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}}.$$

□

Remarks

- The WM algorithm guarantees a bound for m_T,

  $$m_T \leq O(\ln N) + \text{ constant } \cdot |\text{mistakes of the best expert}|.$$

- This bound requires no assumption about the sequence of items and labels generated.
- In the realizable case where $m_T^* = 0$, the bound reduces to $m_T \leq O(\ln N)$ as for the Halving algorithm.

Drawback of Deterministic Algorithms

- In the case of zero-one loss, no deterministic algorithm can achieve a regret $R_T = o(T)$ over all sequences of items.
- For any deterministic algorithm $\mathbb{A}$ and any $t \in [1, T]$, we can adversarially select $c(\omega_t)$ to be 1 if the algorithm predicts 0, and choose it to be 0 otherwise. Thus, $\mathbb{A}$ makes a mistake at every item of such a sequence and its cumulative mistake is $m_T = T$.
- The number m_T^* of mistakes made by the best expert is no greater than $T/2$.
  - Assume for example that $N = 2$ and that one expert always predicts 0, the other one always 1. The error of the best expert over that sequence (and in fact any sequence of that length) is then at most $m_T^* \leq T/2$.

- Thus for any deterministic algorithm, we have

$$R_T = m_T - m_T^* \geq T/2$$

  which shows that $R_T = o(T)$ cannot be achieved in general.

- This leads us to consider randomized algorithms.

The Randomized Weighted Majority Algorithm

RANDOMIZED-WEIGHTED-MAJORITY(N)

1. **for** $i \leftarrow 1$ **to** N **do**
2. $\quad w_{1,i} \leftarrow 1$
3. $\quad p_{1,i} \leftarrow 1/N$
4. **for** $t \leftarrow 1$ **to** T **do**
5. $\quad$ **for** $i \leftarrow 1$ **to** N **do**
6. $\quad\quad$ **if** $(l_{t,i} = 1)$ **then**
7. $\quad\quad\quad w_{t+1,i} \leftarrow \beta w_{t,i}$
8. $\quad\quad$ **else**
9. $\quad\quad\quad w_{t+1,i} \leftarrow w_{t,i}$
10. $\quad W_{t+1} \leftarrow \sum_{i=1}^{N} w_{t+1,i}$

11. **for** $i \leftarrow 1$ **to** N **do**

12. $p_{t+1,i} \leftarrow w_{t+1,i}/W_{t+1}$

13. **return** $\mathbf{w}_{T+1}$

Randomized Scenario of On-Line Learning

- $Q = \{1, \ldots, N\}$: a set of N available actions.
  - For example, action i corresponds to the advice made by expert i.
- $\mathbf{p}_t$: a distribution over the set Q of N actions used by an on-line algorithm $\mathbb{A}$ to select action i to make a prediction $\hat{y}_t$ for the label $c(\omega_t)$ of the item ω_t at each round $t \in [1, T]$.
- $\mathbf{l}_t$: a loss vector received by the on-line algorithm $\mathbb{A}$, whose ith component $l_{t,i} \in \{0, 1\}$ is the zero-one loss associated with action i.
- $L_t = \sum_{i=1}^{N} p_{t,i} l_{t,i}$: the incurred expected loss at the tth round.
  - $L_t \leq 1$.
- $\mathcal{L}_T = \sum_{t=1}^{T} L_t$: the total loss incurred by the on-line algorithm $\mathbb{A}$ over T rounds.

- $\mathcal{L}_{T,i} = \sum_{t=1}^{T} l_{t,i}$: the total loss associated to action i.
- $\mathcal{L}_T^{\min} = \min_{i \in Q} \mathcal{L}_{T,i}$: the minimal loss of a single action.
- The regret R_T of the on-line algorithm after T rounds is also defined by the difference of the total loss of the algorithm and that of the best single action:

$$R_T = \mathcal{L}_T - \mathcal{L}_T^{\min}.$$

- The following theorem gives a strong guarantee on the regret R_T of the randomized weighted majority (RWM) algorithm, showing that it is in $O(\sqrt{T \ln N})$.

Randomized Weighted Majority - Bound for Non-Realizable Concepts

Theorem 7.4: Let

- N : the number of actions;
- $\beta \in [1/2, 1)$: the weight reduction factor.

Then the total loss of the RWM algorithm on any sequence of T items is

$$\mathcal{L}_T \leq \frac{\ln N}{1-\beta} + (2-\beta)\mathcal{L}_T^{\min}.$$

In particular, for $\beta = \max(1/2, 1 - \sqrt{(\ln N)/T})$, the total loss is

$$\mathcal{L}_T \leq \mathcal{L}_T^{\min} + \begin{cases} 2\sqrt{T \ln N}, & \text{if } \sqrt{(\ln N)/T} \leq 1/2, \\ 2\ln N + \frac{T}{2}, & \text{otherwise.} \end{cases}$$

Thus for sufficient large T such that $\sqrt{(\ln N)/T} \leq 1/2$, we have

$$\mathcal{L}_T \leq \mathcal{L}_T^{\min} + 2\sqrt{T \ln N}.$$

Proof.

- $W_t \triangleq \sum_{i=1}^{N} w_{t,i}, t \in [1, T]$: the potential function.
  - $W_1 = N$.
- By the RWM algorithm, for each $t \in [1, T]$, we have

$$\begin{aligned} W_{t+1} &= \sum_{i:l_{t,i}=1} \beta w_{t,i} + \sum_{i:l_{t,i}=0} w_{t,i} \\ &= W_t + (\beta - 1) \sum_{i:l_{t,i}=1} w_{t,i} \\ &= W_t + (\beta - 1) W_t \sum_{i:l_{t,i}=1} p_{t,i} \qquad \text{since } p_{t,i} = \frac{w_{t,i}}{W_t} \\ &= W_t + (\beta - 1) W_t L_t = W_t(1 - (1 - \beta)L_t). \end{aligned}$$

- Since $W_1 = N$, we have

$$W_{T+1} = N \prod_{t=1}^{T} (1 - (1 - \beta) L_t).$$

- $W_{T+1} = \sum_{j=1}^{N} w_{T+1,j} \geq \max_{i \in Q} w_{T+1,i} = \beta^{\mathcal{L}_T^{\min}}$.

Now we have

$$\beta^{\mathcal{L}_T^{\min}} \leq W_{T+1} = N \prod_{t=1}^{T} (1 - (1 - \beta) L_t)$$

and then

$$\mathcal{L}_T^{\min} \ln \beta \leq \ln N + \sum_{t=1}^{T} \ln(1 - (1-\beta)L_t)$$

$$\Rightarrow \quad \mathcal{L}_T^{\min} \ln \beta \leq \ln N - (1-\beta) \sum_{t=1}^{T} L_t \text{ since } \ln(1-x) \leq -x \ \forall \ x < 1$$

$$\Rightarrow \quad \mathcal{L}_T^{\min} \ln \beta \leq \ln N - (1-\beta)\mathcal{L}_T$$

$$\Rightarrow \quad \mathcal{L}_T \leq \frac{\ln N}{1-\beta} - \frac{\ln \beta}{1-\beta} \mathcal{L}_T^{\min}$$

$$\Rightarrow \quad \mathcal{L}_T \leq \frac{\ln N}{1-\beta} - \frac{\ln(1-(1-\beta))}{1-\beta} \mathcal{L}_T^{\min}.$$

Since $-\ln(1-x) \leq x + x^2$ for all $x \in [0, 1/2]$ and $\beta \in [1/2, 1)$, we have

$$-\ln(1-(1-\beta)) \leq (1-\beta) + (1-\beta)^2 = (1-\beta)(2-\beta)$$

and then

$$\mathcal{L}_T \le \frac{\ln N}{1-\beta} + (2-\beta)\mathcal{L}_T^{\min}.$$

Since $\mathcal{L}_T^{\min} \le T$, we have

$$\mathcal{L}_T \le \mathcal{L}_T^{\min} + \frac{\ln N}{1-\beta} + (1-\beta)T.$$

The upper bound in above is minimized with
$\beta = \beta_0 = 1 - \sqrt{(\ln N)/T}$ if $\sqrt{(\ln N)/T} \le 1/2$ and with
$\beta = \beta_0 = 1/2$ if $\sqrt{(\ln N)/T} > 1/2$. With $\beta = \beta_0$, we have

$$\mathcal{L}_T \le \mathcal{L}_T^{\min} + \begin{cases} 2\sqrt{T \ln N}, & \text{if } \sqrt{(\ln N)/T} \le 1/2, \\ 2\ln N + \frac{T}{2}, & \text{otherwise.} \end{cases}$$

□

Remarks

- The RWM algorithm has the average regret or regret per round R_T/T decreases as $O(1/\sqrt{T})$.
- These results are optimal since for $T \geq N$, a lower bound of $R_T = \Omega(\sqrt{T \ln N})$ can be proven for any algorithm or from a result shown in the following theorem.

Khintchine-Kahane Inequality

Let

- $\sigma = (\sigma_1, \ldots, \sigma_m)$: Rademacher variables, i.e., i.i.d. random variables taking values in $\{-1, +1\}$ with equal probability.
- $\mathbf{x}_i, \ldots, \mathbf{x}_m$: vectors in a normed vector space $(\mathbb{H}, \|\cdot\|)$.

Then we have

$$\frac{1}{2} \underset{\sigma}{E} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|^2 \right] \leq \left(\underset{\sigma}{E} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\| \right] \right)^2 \leq \underset{\sigma}{E} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|^2 \right].$$

A Lower Bound of Regret for Randomized Algorithms

Theorem 7.5: Let

- $N = 2$.

There exists a stochastic sequence of losses for which the regret of any random on-line learning algorithm has $E[R_T] \geq \sqrt{T/8}$.

Proof.

- For each $t \in [1, T]$, the loss vector $\mathbf{l}_t$ takes the values $\mathbf{l}_{01} = (0, 1)^T$ and $\mathbf{l}_{10} = (1, 0)^T$ with equal probability.
- The expected total loss of any random on-line algorithm is

$$E[\mathcal{L}_T] = E\left[\sum_{t=1}^{T} \mathbf{p}_t \cdot \mathbf{l}_t\right] = \sum_{t=1}^{T} \mathbf{p}_t \cdot E[\mathbf{l}_t] = \sum_{t=1}^{T} \frac{1}{2} p_{t,1} + \frac{1}{2}(1 - p_{t,1}) = \frac{T}{2},$$

  where $\mathbf{p}_t$ is the distribution selected by the random on-line

algorithm at round t.

- $\mathcal{L}_{T,1} + \mathcal{L}_{T,2} = T$.
- $\mathcal{L}_T^{\min} = \min(\mathcal{L}_{T,1}, \mathcal{L}_{T,2}) = \frac{1}{2}(\mathcal{L}_{T,1} + \mathcal{L}_{T,2} - |\mathcal{L}_{T,1} - \mathcal{L}_{T,2}|) = \frac{T}{2} - |\mathcal{L}_{T,1} - \frac{T}{2}|$.
- The expected regret of the random on-line algorithm is

$$E[R_T] = E[\mathcal{L}_T] - E[\mathcal{L}_T^{\min}] = E[|\mathcal{L}_{T,1} - \frac{T}{2}|].$$

- $\sigma_t, t \in [1, T]$: Rademacher variables taking values in $\{-1, +1\}$ with equal probability.
- $\mathcal{L}_{T,1} = \sum_{t=1}^{T} \frac{1+\sigma_t}{2} = \frac{T}{2} + \frac{1}{2}\sum_{t+1}^{T} \sigma_t$.
- With $\alpha_t = 1/2$ for all $t \in [1, T]$, we have

$$E[R_T] = E\left[\left|\sum_{t=1}^{T} \sigma_t \alpha_t\right|\right] \geq \sqrt{\frac{1}{2}\left|\sum_{t=1}^{T} \sigma_t \alpha_t\right|^2} = \sqrt{\frac{1}{2}\sum_{t=1}^{T} \alpha_t^2} = \sqrt{\frac{T}{8}}$$

by the Khintchine-Kahane inequality and the fact that $E[\sigma_i \sigma_j] = 0$ for all $i \neq j$. $\square$

The Exponential Weighted Average Algorithm

EXPONENTIAL-WEIGHTED-AVERAGE(N)

1. **for** $i \leftarrow 1$ **to** N **do**
2. $\quad w_{1,i} \leftarrow 1$
3. **for** $t \leftarrow 1$ **to** T **do**
4. $\quad$ RECEIVE(ω_t)
5. $\quad \hat{y}_t \leftarrow \frac{\sum_{i=1}^{N} w_{t,i}\hat{y}_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$
6. $\quad$ RECEIVE($c(\omega_t)$)
7. $\quad$ **for** $i \leftarrow 1$ **to** N **do**
8. $\quad\quad w_{t+1,i} \leftarrow w_{t,i} e^{-\eta L(\hat{y}_{t,i}, c(\omega_t))}$
9. **return** $\mathbf{w}_{T+1}$

Remarks

- The WM algorithm can be extended to other loss functions L taking values in $[0, 1]$.
- For the exponential weighted average (EWA) algorithm presented here, we assume that the loss function $L(y', y)$ is convex in its first argument y'.
- Although the EWA algorithm is deterministic, it admits a very favorable regret guarantee, as shown in the next theorem.
- The EWA algorithm's prediction is the weighted average of the advice of N experts,

$$\hat{y}_t = \frac{\sum_{i=1}^{N} w_{t,i} \hat{y}_{t,i}}{\sum_{i=1}^{N} w_{t,i}}.$$

- The EWA algorithm updates the weights at the end of round t

according to the following rule:

$$w_{t+1,i} = w_{t,i} e^{-\eta L(\hat{y}_{t,i}, c(\omega_t))} = e^{-\eta L_{t,i}},$$

where $L_{t,i} = \sum_{s=1}^{t} L(\hat{y}_{s,i}, c(\omega_s))$ is the total loss incurred by expert i after t rounds.

Hoeffding's Lemma

Let

- X : a r.v. with zero mean, i.e., $E[X] = 0$.
- $a \leq X \leq b$ with $b > a$.

Then for any $t > 0$, we have

$$E[e^{tX}] \leq e^{\frac{t^2(b-a)^2}{8}}.$$

A Regret Bound for the EWA Algorithm

Theorem 7.6: Let

- $\{1, 2, \ldots, N\}$: the set of N experts.
- $L(y', y)$: a loss function which is convex in the first argument y' and takes values in $[0, 1]$;
- $\eta > 0$: a weight update parameter.

Then, for any label sequence $c(\omega_1), \ldots, c(\omega_T)$, the regret of the exponential weighted average (EWA) algorithm after T rounds has

$$R_T \leq \frac{\ln N}{\eta} + \frac{\eta T}{8}.$$

In particular, for $\eta = \sqrt{8(\ln N)/T}$, the regret is bounded as

$$R_T \leq \sqrt{(T/2) \ln N}.$$

Proof.

- $\Phi_t \triangleq \ln\left(\sum_{i=1}^{N} w_{t,i}\right), t \in [1,T]$: the potential function.
  - $\Phi_1 = \ln N$.
- $\mathbf{p}_t, t \in [1,T]$: the distribution over $\{1,2,\ldots,N\}$ with $p_{t,i} = \frac{w_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$.
- By the EWA algorithm, the difference of two consecutive potential values is

$$\Phi_{t+1} - \Phi_t = \ln \frac{\sum_{i=1}^{N} w_{t,i} e^{-\eta L(\hat{y}_{t,i}, c(\omega_t))}}{\sum_{i=1}^{N} w_{t,i}} = \ln \underset{\mathbf{p}_t}{E}[e^{\eta X}],$$

  where $X(i) = -L(\hat{y}_{t,i}, c(\omega_t)) \in [-1,0]$ is a r.v. on $[1,N]$ with distribution $\mathbf{p}_t$.
- By applying Hoeffding's lemma to the centered r.v. $X - \underset{\mathbf{p}_t}{E}[X]$,

we have

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \ln \underset{\mathbf{p}_t}{E}[e^{\eta(X - \underset{\mathbf{p}_t}{E}[X]) + \eta \underset{\mathbf{p}_t}{E}[X]}] \\
&\leq \frac{\eta^2}{8} + \eta \underset{\mathbf{p}_t}{E}[X] \\
&= \frac{\eta^2}{8} - \eta \underset{\mathbf{p}_t}{E}[L(\hat{y}_{t,i}, c(\omega_t))] \\
&\leq \frac{\eta^2}{8} - \eta L(\underset{\mathbf{p}_t}{E}[\hat{y}_{t,i}], c(\omega_t)) \\
&\quad \text{by the convexity of } L \text{ in the first argument} \\
&= \frac{\eta^2}{8} - \eta L(\hat{y}, c(\omega_t)).
\end{aligned}$$

- Summing up for $t \in [1, T]$, we have an upper bound of the

potential function

$$\Phi_{T+1} - \Phi_1 \leq \frac{\eta^2 T}{8} - \eta \sum_{t=1}^{T} L(\hat{y}, c(\omega_t)).$$

- A lower bound can be obtained as

$$\begin{aligned} \Phi_{T+1} - \Phi_1 &= \ln \sum_{i=1}^{N} e^{-\eta L_{T,i}} - \ln N \\ &\geq \ln \max_{i \in [1,N]} e^{-\eta L_{T,i}} - \ln N \\ &= -\eta \min_{i \in [1,N]} L_{T,i} - \ln N. \end{aligned}$$

Now we have

$$-\eta \min_{i \in [1,N]} L_{T,i} - \ln N \leq \frac{\eta^2 T}{8} - \eta \sum_{t=1}^{T} L(\hat{y}, c(\omega_t))$$

which implies that

$$R_T = \sum_{t=1}^{T} L(\hat{y}, c(\omega_t)) - \min_{i \in [1,N]} L_{T,i} \leq \frac{\ln N}{\eta} + \frac{\eta T}{8}.$$

The upper bound in above is minimized with $\eta = \sqrt{8(\ln N)/T}$ and we have

$$R_T \leq \sqrt{(T/2) \ln N}.$$

□

Remarks

- The optimal choice of η in Theorem 7.6 requires knowledge of the horizon T, which is an apparent disadvantage of this analysis.
- However, we can use a standard doubling trick to eliminate this requirement, at the price of a small constant factor.
- The standard doubling trick : dividing time into periods $[2^k, 2^{k+1} - 1]$ of length 2^k with $k = 0, \ldots, n$.
- Assume that $2^n \leq T \leq 2^{n+1} - 1$.
- Choose $\eta_k = \sqrt{\frac{8 \ln N}{2^k}}$ in each period $[2^k, 2^{k+1} - 1]$ of length 2^k with $k = 0, \ldots, n$.

A Regret Bound for the EWA Algorithm with Doubling Trick

Theorem 7.7: Let

- $\{1, 2, \ldots, N\}$: the set of N experts.
- $L(y', y)$: a loss function which is convex in the first argument y' and takes values in $[0, 1]$.

Then, for any label sequence $c(\omega_1), \ldots, c(\omega_T)$, the regret of the exponential weighted average (EWA) algorithm after T rounds has

$$R_T \leq \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{(T/2)\ln N} - \frac{1}{\sqrt{2}-1}\sqrt{(\ln N)/2}.$$

Proof.

- $n \triangleq \lfloor \log_2 T \rfloor$.
- $I_k \triangleq [2^k, 2^{k+1} - 1], k \in [0, n]$.
- $\mathcal{L}_{I_k}$: the total loss incurred in the interval I_k.
- By Theorem 7.6, for any $k \in [0, n]$, we have

$$\mathcal{L}_{I_k} - \min_{i \in [1,N]} \mathcal{L}_{I_k,i} \leq \sqrt{(2^k/2) \ln N}.$$

Thus we have

$$\begin{aligned}
\mathcal{L}_T \le \sum_{k=0}^{n} \mathcal{L}_{I_k} &\le \sum_{k=0}^{n} \min_{i\in[1,N]} \mathcal{L}_{I_k,i} + \sum_{k=0}^{n} \sqrt{2^k (\ln N)/2} \\
&\le \min_{i\in[1,N]} \sum_{k=0}^{n} \mathcal{L}_{I_k,i} + \sqrt{(\ln N)/2} \sum_{k=0}^{n} 2^{k/2} \\
&= \min_{i\in[1,N]} \mathcal{L}_{T,i} + \sqrt{(\ln N)/2}\ \frac{2^{(n+1)/2}-1}{\sqrt{2}-1}.
\end{aligned}$$

Since $2^n \le T$, we have

$$\frac{2^{(n+1)/2}-1}{\sqrt{2}-1} \le \frac{\sqrt{2T}-1}{\sqrt{2}-1} \le \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{T} - \frac{1}{\sqrt{2}-1}$$

so that

$$R_T = \mathcal{L}_T - \min_{i\in[1,N]} \mathcal{L}_{T,i} \le \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{(T/2)\ln N} - \frac{1}{\sqrt{2}-1}\sqrt{(\ln N)/2}.$$

□