E1 245: Online Prediction & Learning **Fall 2014**

Lecture 2 — August 7

Lecturer: Aditya Gopalan *Scribe: Indu John*

2.1 Recap

In the last class, we described the MAJORITY algorithm(*MAJ*) for 1-bit prediction and established its mistake bound, under the assumption that there exists a perfect expert who makes no mistakes.

Today we will see a more general algorithm namely WEIGHTED MAJORITY which makes no such assumptions. We also introduce the general model of *Prediction with expert advice* and define a notion of performance called *Regret* for any algorithm in this setup. Finally, we look at the EXPONENTIAL WEIGHTS prediction algorithm which can be shown to perform well with respect to regret minimization.

Exercise : We showed that the MAJORITY algorithm makes at most $\log_2 N$ mistakes using the advice of N experts whenever the best expert makes no mistakes. Show that a straightforward modification of MAJORITY makes at most $O((m+1)\log_2 N)$ mistakes when the best expert makes $m \geq 0$ mistakes.

We want to get rid of any assumptions on the number of mistakes made by the experts. Also, we would like to bring down the upper bound on number of mistakes made by the algorithm. Let's look at the WEIGHTED MAJORITY algorithm that achieves both.

2.2 Weighted majority algorithm

Algorithm 1 Weighted Majority($W-MAJ$)

Parameter: $\varepsilon \in [0,1]$
Initialize : the weight for expert i, $w_{i,0} = 1 \quad \forall i$
for $t = 1,2,3,...$ **do**
    Predict

$$\hat{p}_t = \begin{cases} 1 & \text{if } \sum_{i:f_{i,t}=1} w_{i,t-1} \geq \sum_{i:f_{i,t}=0} w_{i,t-1} \\ 0 & \text{otherwise} \end{cases}$$

    Observe y_t
    $w_{i,t} = w_{i,t-1}(1-\varepsilon)^{\mathbb{I}\{f_{i,t} \neq y_t\}} \quad \forall i$

Note that when $\varepsilon = 1$, this algorithm becomes the same as MAJORITY. When an expert makes a wrong prediction, his weight is set to 0(equivalent to throwing him out).

Theorem 2.1 (Mistake bound for $W-MAJ$). *Let $\varepsilon \in \left[0, \frac{1}{2}\right]$. Then, for each expert i,*

$$M_T(W-MAJ) \leq 2(1+\varepsilon)M_T(i) + 2\frac{logN}{\varepsilon}$$

where $M_T(W-MAJ)$ is the number of mistakes of $W-MAJ$ upto time T, $M_T(i)$ is the number of mistakes of expert i upto time T.
This implies,

$$M_T(W-MAJ) \leq 2(1+\varepsilon)\min_{i=1}^{N} M_T(i) + 2\frac{logN}{\varepsilon}$$

Proof: We will use an argument using potential functions to prove the theorem. Define,

$$\Phi_t := \sum_{i=1}^{N} w_{i,t} \quad \text{for } t = 1,2,...,T.$$

We will track the evolution of Φ_t with t.
At the beginning, we have, $\Phi_0 = N$.
Note that,

(1) $\Phi_T \geq w_{i,T} = w_{i,0}(1-\varepsilon)^{M_T(i)} = 1(1-\varepsilon)^{M_T(i)}$

(2) Suppose $W-MAJ$ predicts wrongly at time t. Then, at least half the total weight goes down by $(1-\varepsilon)$. Therefore,

$$\begin{aligned} \Phi_{t+1} &\leq \frac{\Phi_t}{2} + \frac{\Phi_t}{2}(1-\varepsilon) \\ &= \Phi_t\left(1-\frac{\varepsilon}{2}\right) \\ \Rightarrow \Phi_T &\leq \Phi_0\left(1-\frac{\varepsilon}{2}\right)^{M_T(W-MAJ)} = N\left(1-\frac{\varepsilon}{2}\right)^{M_T(W-MAJ)} \end{aligned}$$

From (1) and (2),

$$\begin{aligned} N\left(1-\frac{\varepsilon}{2}\right)^{M_T(W-MAJ)} &\geq (1-\varepsilon)^{M_T(i)} \\ \Rightarrow -logN + M_T(W-MAJ)log\left(\frac{1}{1-\frac{\varepsilon}{2}}\right) &\leq M_T(i)log\left(\frac{1}{1-\varepsilon}\right) \\ \Rightarrow M_T(W-MAJ) &\leq M_T(i)\left(\frac{log\frac{1}{1-\varepsilon}}{log\frac{1}{1-\frac{\varepsilon}{2}}}\right) + \frac{logN}{log\left(\frac{1}{1-\frac{\varepsilon}{2}}\right)} \end{aligned}$$

Using the inequalities

$$\begin{aligned} log(1+x) &\leq x \quad \forall x \geq 0, \\ -log(1-x) &\leq x + x^2 \quad \forall x \in \left[0, \frac{1}{2}\right], \end{aligned}$$

we obtain

$$M_T(W-MAJ) \leq 2(1+\varepsilon)M_T(i) + \frac{2}{\varepsilon} logN$$

□

Note on tuning the parameter ε: It is easy to see that the optimal ε which gives the tighest upper bound is $\sqrt{\frac{logN}{M_T(i)}}$. With this ε, the bound becomes $\mathbf{2}M_T(i) + 4\sqrt{M_T(i)logN}$.
In general, the **2** multiplying $M_T(i)$ can't be eliminated without introducing randomness in the algorithm.

2.3 Prediction with expert advice

The general model for prediction with expert advice is the following. We have

- a Decision space $\mathscr{D}$,
- an Outcome space $\mathscr{Y}$,
- a Loss function $l : \mathscr{D} \times \mathscr{Y} \to \mathbb{R}$
- the set of experts $\mathscr{E}$

At each round $t = 1, 2, ...,$

- Environment picks $y_t \in \mathscr{Y}$
- Algorithm receives expert advice $(f_{i,t})_{i=1}^N, \quad f_{i,t} \in \mathscr{D}$
- Algorithm predicts $\hat{p}_t \in \mathscr{D}$
- Algorithm sees y_t
- Algorithm suffers loss for current round : $l(\hat{p}_t, y_t)$

Examples:

(1) 1-bit prediction
$\mathscr{D} = \mathscr{Y} = \{0,1\}$
$l(p,y) = \mathbb{I}\{p \neq y\}$ ("0-1" loss)

(2) Online regression
Here we want to fit a function to observed data.
$\mathscr{X} \subseteq \mathbb{R}^d$
$\mathscr{D}$ = some class of functions $g : \mathbb{R}^d \to \mathbb{R}$. eg., linear functions, polynomials of degree 5
$\mathscr{E} = \mathscr{D}$
$\mathscr{Y} = \mathbb{R}$

- Environment picks ($x_t \in \mathscr{X}, y_t \in \mathscr{Y}$)
- x_t is revealed to algorithm
- Each expert $g \in \mathscr{E}$ recommends $g(x_t)$
- Algorithm predicts $\hat{g}_t \in \mathscr{D}$
- Algorithm suffers loss $l(\hat{g}_t, (x_t, y_t))$. eg. in the case of least squares regression, $l(\hat{g}_t, (x_t, y_t)) = (\hat{g}_t(x_t) - y_t)^2$

The goal of the algorithm is to minimize the regret

$$\begin{aligned} R_t &= \max_{i \in \mathscr{E}} R_{i,t}, \\ \text{where } R_{i,t} &:= \sum_{s=1}^{t} l(\hat{p}_s, y_s) - \sum_{s=1}^{t} l(f_{i,s}, y_s) \end{aligned}$$

The regret can alternatively be expressed as

$$R_t = \sum_{s=1}^{t} l(\hat{p}_s, y_s) - \min_{i \in \mathscr{E}} \sum_{s=1}^{t} l(f_{i,s}, y_s),$$

the difference between the cumulative loss of the algorithm and the cumulative loss of the best expert.

It would be nice to have $R_t = o(t)$, or equivalently, $\frac{R_t}{t} \to 0$ as $t \to \infty$. In other words, the algorithm 'learns' the best strategy over time. This property is known as *"Hannan consistency"*.

We now introduce the EXPONENTIAL WEIGHTS algorithm which performs very well with respect to regret minimization when the loss functions are convex.

2.4 Exponential weights algorithm

Let the decisions and losses be convex. i.e., $\mathscr{D}$ is a convex set and $l : \mathscr{D} \times \mathscr{Y} \to \mathbb{R}$ is convex on $\mathscr{D}$.

Algorithm 2 Exponential Weights($EXP-WTS$)

Parameter : $\eta > 0$ (called learning rate)
Initialize : the weight for expert i, $w_{i,0} = 1 \quad \forall i$
for $t = 1, 2, 3, \ldots$ **do**
    $w_{i,t} = e^{-\eta \sum_{s=1}^{t-1} l(f_{i,s}, y_s)}$
    Predict $\hat{p}_t = \frac{\sum_{i \in \mathscr{E}} f_{i,t} w_{i,t}}{\sum_{i \in \mathscr{E}} w_{i,t}}$ ($\in \mathscr{D}$ since $\mathscr{D}$ is convex).

It may be noted that $e^{-\eta}$ in this algorithm plays the role of $(1-\varepsilon)$ in the WEIGHTED MAJORITY algorithm.

We will derive the regret bound for this algorithm in the next class.

References

[1] Nicolo Cesa-Bianchi and Gabor Lugosi, *Prediction, Learning and Games, Chapter 2, Section 2.1*. Cambridge University Press, 2006.