E1 245: Online Prediction & Learning **Fall 2015**

Lecture 8 — August 27

Lecturer: Aditya Gopalan *Scribe: Mukund Seethamraju*

8.1 MINIMAX REGRET for general CONVEX Prediction Game:

$y = \{0,1\} \quad \mathbb{D} = [0,1] \quad l = |p-y|$
In the last class we have seen the following:

$$\sup_{T,N} \frac{V_T^{(N)}}{\sqrt{\frac{T}{2}\log N}} \geq 1, \qquad V_T^{(N)} = \inf_{algA} \sup_{\{y\}\{t\}|\varepsilon|=N} R_T^A$$

8.1.1 TRACKING EXPERSTS/ COMPETING WITH SHIFTING EXPERTS:

Motivation: So far, our performance have always been measure with respect to the performance of the best SINGLE expert.
e.g. $|\varepsilon| = \mathrm{N} \quad L : \mathbb{D} \times y \to [0,1]$. R-EXP weights algorithm enjoys

$$\mathbb{E}[\sum_{t=1}^{T} l(f_{I_t,t}, y_t)] \leq \min_{i\in[N]} \sum_{t=1}^{T} l(f_{I_t,t}, y_t) + \sqrt{\frac{T}{2}\log \mathrm{N}}$$

In other words, R-EXPWTS is COMPETETIVE with the set of all strategies of the form "pick an expert and follow it for all the time."
But: Then there might be strategies that can switch across experts that can achieve much better cumulative loss.
e.g. Consider $\mathbb{D} = y = \{0,1\} \quad l(p,y) = \mathbb{1}\{p \neq y\}$ & consider the outcome sequence.

$$(y_1, y_2, ...y_T) = (\underbrace{0,0,0,......0}_{\text{k times}}, \underbrace{1,1,1,....1}_{\text{T - k times}})$$

assume k $\approx \frac{\mathrm{T}}{2}$
ε_1 = {always predict '0', always predict '1'}

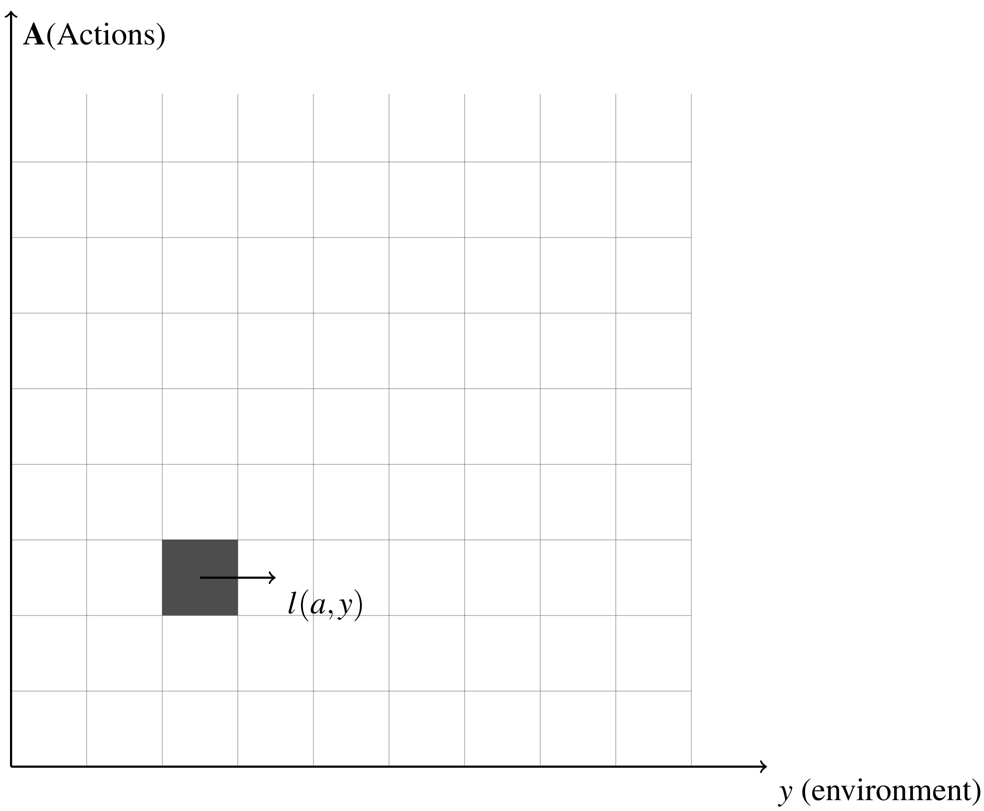
Best possible loss of any expert from ε_1 on $(y_1, y_2, ...y_T)$ is $\gtrapprox \frac{\mathrm{T}}{2}$

ε_2 = {start with predicting '0' (or '1') and toggle atmost once at some time t $\leq$ T}

Best possible CUMULATIVE loss of any 'expert' from ε_2 with repsect to sequences of the form $(y_1, y_2, ...y_T) = 0$

QUESTION: Design predicting strategies that are competitive w.r.t classes of the DYNAMIC/SWITCHING strategies? TRACKING PROBLEM.

Consider the following (randomized) prediction game:
This is a Actions game, essentially same PREDICTION with EXPERT ADVICE.
Set of actions $\mathbf{A} = \{1,2,3.... N\}$
set of outcomes : y
loss function: $l : \mathbf{A} \times y \rightarrow [0,1]$
If y is finite:



In the above figure set of actions = (1,2,3,..... ∞) and set of outcomes(y) is finite.
At each time t $\geq$ 1:

- Learner observes a distribution $\hat{p}_t \in \Delta_n$
- Environment chooses $y_t \in y$
- Learner sets y_t, suffers (expected) loss $\sum_{i=1}^{n} \hat{p}_{ti} l(i, y_t)$

$R_T^{(0)}$ Regret of the learner over T rounds = $\sum_{t=1}^{T} \sum_{i=1}^{n} \hat{p}_{ti} l(i, y_t) - \min_{i \in [n]} \sum_{t=1}^{T} l(i, y_t)$

Previously we showed that if the learner uses R_EXPWTS (η) over ($\mathbf{A}$) then $R_T^{(0)} \leq \frac{\log N}{\eta} + \frac{\eta T}{8}$

Definition(TRACKING REGRET):

Fix a time horizon $\mathrm{T} \in \mathbb{N}$

Let $(i_1, i_2, i_3, ...i_T) \in [N]^{\mathrm{T}} = \mathbb{A}^{\mathrm{T}}$ by a length 'T' sequence of actions.

The tracking regret of an algorithm $\mathbb{A}$ w.r.t $(i_1, i_2, i_3, ...i_T)$ is $\sum_{t=1}^{T} \sum_{i=1}^{n} \hat{p_{ti}} l(i, y_t) - \sum_{t=1}^{T} l(i_t, y_t)$

Definition: For $(i_1, i_2, i_3, ...i_T) \in [N]^{\mathrm{T}}$, Let $\#(i_1, i_2, i_3, ...i_T) = \sum_{t=1}^{\mathrm{T}} \mathbb{1}\{i_t \neq i_{t-1}\}$

Definition: Let $\mathscr{E} \subseteq [N]^{\mathrm{T}}$. The traccking regret of an algorithm $\mathbb{A}$ w.r.t $\mathscr{E}$ is

$$\mathbb{R}_{\mathrm{T}}(\mathscr{E}) = \max_{e \in \mathscr{E}} R_{\mathrm{T}}(\mathscr{E}) = \sum_{t=1}^{T} \sum_{i=1}^{n} \hat{p_{ti}} l(i, y_t) - \min_{e \in \mathscr{E}} \sum_{t=1}^{T} l(e_t, y_t) \quad \text{where } e = (e_1, e_2, ...e_{\mathrm{T}})$$

NOTE: Define $\mathscr{E}_m := \{(i_1, i_2, ...i_{\mathrm{T}}) \in [N]^{\mathrm{T}} : \#(i_1, i_2, ...i_{\mathrm{T}}) \leq m\}$

$R_{\mathrm{T}}(\mathscr{E}_m)$ is the usual notation of expected regret so far.

IDEA: WE can just apply RANDOMIZED EXPWTS algorithm by treating each compound action $(i_1, i_2, ...i_{\mathrm{T}})$ in $\mathscr{E}$ as a "new action".

$$(i_1^{(1)}, i_2^{(1)}, i_3^{(1)}, ...i_{\mathrm{T}}^{(1)}) \rightarrow \text{Weight} = 1$$

$$(i_1^{(2)}, i_2^{(2)}, i_3^{(2)}, ...i_{\mathrm{T}}^{(2)}) \rightarrow \text{Weight} = 1$$

$$(i_1^{(3)}, i_2^{(3)}, i_3^{(3)}, ...i_{\mathrm{T}}^{(3)}) \rightarrow \text{Weight} = 1$$

$$\vdots$$

$$(i_1^{|\mathscr{E}|}, i_2^{|\mathscr{E}|}, i_3^{|\mathscr{E}|}, ...i_{\mathrm{T}}^{|\mathscr{E}|}) \rightarrow \text{Weight} = 1$$

$\rightarrow \forall\ (y_1, y_2, ...y_{\mathrm{T}}) \in \mathbf{y}^{\mathrm{T}} \quad \mathbb{R}_{\mathrm{T}}(\mathscr{E}) \leq \sqrt{\frac{\mathrm{T}}{2} \log |\mathscr{E}|}$

NOTE: If $\mathscr{E} \equiv \mathscr{E}_{\mathrm{T}-1} \equiv [N]^{\mathrm{T}}$ then

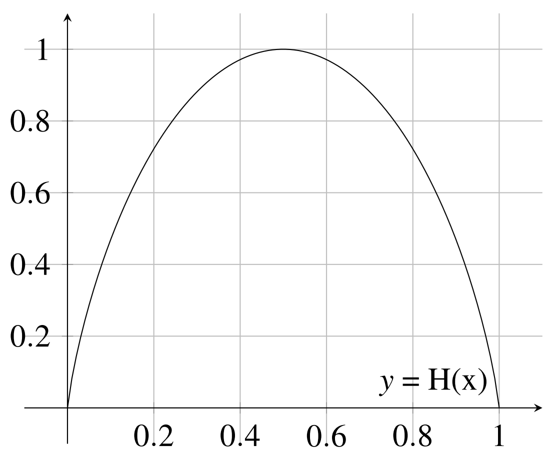
$$\mathbb{R}_{\mathrm{T}}(\mathscr{E}) = \sum_{t=1}^{T} \sum_{i=1}^{n} \hat{p_{ti}} l(i, y_t) - \sum_{t=1}^{T} \min_{i \in [N]} l(i, t) \leq \sqrt{\frac{\mathrm{T}}{2} \log N^{\mathrm{T}}} = \mathrm{T} \sqrt{\frac{\log N}{2}}$$

Infact no(casual) algorithm gets $\mathbb{R}_{\mathrm{T}}(\mathscr{E}_{(\mathrm{T}-1)})$ sublinear in T.

Lemma: $|\mathscr{E}_{(\mathrm{m})}| = \sum_{k=0}^{\mathrm{m}} \binom{\mathrm{T}-1}{k} N(N-1)^K$

Proof: $\binom{\mathrm{T}-1}{k} N(N-1)^K =$ # of sequences $(i_1, i_2, ...i_{\mathrm{T}})$ with exactly k switches.

Lemma: $|\mathscr{E}_{\mathrm{m}}| \leq N^{\mathrm{m}+1} \mathrm{e}^{(\mathrm{T}-1)\mathrm{H}(\frac{\mathrm{m}}{\mathrm{m}-1})}$
H is the binary ENTROPY function.
H(x) = -[logx + (1 - x)log(1 - x)] $\forall$ x $\in (0,1)$



References:

- Online learning CMPUT 654 by Gabor Bartok, chapter-6
- Prediction Learning and Games by Cesa Bianchi - chapter 5