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

Lecture 9 — September 1

Lecturer: Aditya Gopalan *Scribe: Shreyas S*

9.1 Recap - Tracking the Best Expert

In the previous lecture, we introduced another prediction game setting called **Actions Game**. The setting is a tuple $(\mathbb{A}, \mathbb{Y}, \ell)$, where

1. $\mathbb{A}$ is the set actions of available, given by $\{1, 2, \cdots N\}$.
2. $\mathbb{Y}$ is the space of outcomes.
3. $\ell : \mathbb{A} \times \mathbb{Y} \to \mathbb{R}$ is the loss function.

The game is defined for each time $t \geq 1$ as follows:

1. The predictor plays an (randomized) action i.
2. The predictor sees the outcome y_t an suffers a loss $\ell(i, y_t)$.

The regret of an algorithm $\mathscr{A}$ in actions game setting with respect to a compound-action/ switching-experts i.e., $(i_1, i_2, \cdots, \imath_T) \in [N]^T$ for T rounds is given by

$$\mathbf{R}_T^{\mathscr{A}}(i_1, i_2, \cdots, i_T) = E\left[\sum_{t=1}^{T} \ell(\mathbb{I}_t, y_t) - \sum_{t=1}^{T} \ell(i_t, y_t)\right]$$

where $\mathbb{I}_t$ is the action chosen $\mathscr{A}$ at time t. Recall the definition of space of experts who can switch at-most m times i.e.,

$$\mathscr{E}_{(m)} = \{(i_1, i_2, \cdots, i_T) \in [N]^T : \#(i_1, i_2, \cdots, i_T) \leq m\}$$

where $\#(i_1, i_2, \cdots, i_T) = \sum_{t=1}^{T} 1\{i_{t-1} \neq i_t\}$. Then we proceeded to give definition of the regret of the algorithm $\mathscr{A}$ over $\mathscr{E}_{(m)}$.

$$\mathbf{R}_T^{\mathscr{A}}\left(\mathscr{E}_{(m)}\right) = E\left[\sum_{t=1}^{T} \ell(\mathbb{I}_t, y_t)\right] - \min_{(i_1, i_2, \cdots, i_T) \in \mathscr{E}_{(m)}} \sum_{t=1}^{T} \ell(i_t, y_t)$$

9.2 Bound on $|\mathscr{E}_{(m)}|$

Lemma 9.1.

$$|\mathscr{E}_{(m)}| \ \leq \ N^{m+1} \exp\left[(T-1)H(\frac{m}{(T-1)})\right]$$

when $\forall x \in [0,1]$ *and where* $H(x) = -[x\log x + (1-x)\log(1-x)]$.

Proof: We use the following lemma to prove the bound.

Lemma 9.2. $\forall k$*, such that* $0 \leq k \leq m$*, then* $\binom{n}{k} \leq \exp\left[nH(\frac{k}{n})\right]$.

From the previous lecture we know that,

$$\begin{aligned}|\mathscr{E}_{(m)}| &= \sum_{k=0}^{m} \binom{T-1}{k} N(N-1)^k \\ &\leq \binom{T-1}{m} NN^m, \text{ (By over counting.)} \\ &\leq \exp\left[(T-1)H(\frac{m}{(T-1)})\right] \ N^{m+1}, \text{ (from Lemma (9.2))}\end{aligned}$$

□

9.3 R-EXPWTS Algo over $\mathscr{E}_{(m)}$

We now use R-EXPWTS on the space of switching-experts/compound actions. The R-EXPWTS is run over $\mathscr{E} \subseteq [N]^T$ as follows:

1. Initially $\forall (i_1, \cdots, i_T) \in \mathscr{E}$, we set $w'_1(i_1, \cdots, i_T) = 1$.
2. For each time $t \geq 1$, sample $E_t \in \mathscr{E}$, according to current weights w'_t. Suppose $E_t = (i_1, \cdots, i_T)$ then play the action $I_t = i_t$.
3. $\forall (j_1, j_2, \cdots, j_T) \in \mathscr{E}$, update weights

$$w'_{t+1}(j_1, \cdots, j_T) = w'_t(j_1, \cdots, j_T) exp\left[-\eta \ell(i_t, y_t)\right]$$

Note: For $\mathscr{E} = \mathscr{E}_{(0)}$, we get back R-EXPWTS($\eta$) over N actions.

Note: For $T >> 1$, $\mathscr{E} = \mathscr{E}_{(m)}$, $m < \frac{T}{2}$,

$$\begin{aligned}|\mathscr{E}_{(m)}| &= \sum_{k=0}^{m} \binom{T-1}{k} N(N-1)^k \\ &\geq \binom{T-1}{m} (N-1)^m \\ &\geq \left(\frac{T-1}{m}\right)^m (N-1)^m.\end{aligned}$$

Theorem 9.3. *The R-EXPWTS algorithm when run with $\mathscr{E}_{(m)}$ as the set of "experts", then*

$$E\left[\sum_{t=1}^{T}\ell(\mathbb{I}_t, y_t)\right] \le \min_{(i_1,i_2,\cdots,i_T)\in\mathscr{E}_{(m)}} \sum_{t=1}^{T}\ell(i_t,y_t) + \sqrt{\frac{T}{2}\left[(m+1)\log N + (T-1)H\left(\frac{m}{T-1}\right)\right]}$$

Note:

$$\begin{aligned}
&\text{If } \sqrt{\frac{T}{2}\left[(m+1)\log N + (T-1)H\left(\frac{m}{T-1}\right)\right]} = O(T) \\
&\Leftrightarrow \lim_{T\to\infty}\sqrt{\left[\frac{(m+1)\log N}{T} + H\left(\frac{m}{T}\right)\right]} = 0 \\
&\Leftrightarrow \frac{m}{T} \leftarrow 0 \text{ as } T \to \infty \\
&m = O(T)
\end{aligned}$$

Naively running R-EXPWTS over $\mathscr{E}_{(m)}$ "experts" is impossible in practice. But we'll show an efficient algorithm that updates N weights. Before that, lets look at the regret of "standard R-EXPWTS" with non uniform initial weights.

9.3.1 R-EXPWTS Algo with non uniform initial weights

We begin the section by stating a lemma (without proof).

Lemma 9.4. *Let us consider an N actions game with losses $\in [0,1]$ and initialize the R-EXPWTS algorithm with initial weights $(w_{1,1}, w_{2,1}, \cdots, w_{N,1}) \in \Delta_N$. Then*

$$\sum_{t=1}^{T}\sum_{t=1}^{N} p_{ti}\ell(i,y_t) \le \frac{1}{\eta} log \frac{1}{W_{T+1}} + \frac{\eta T}{8}$$

where $W_{T+1} = \sum_{t=1}^{N} w_{i,T+1} = \sum_{t=1}^{N} w_{i,1} \exp\left(\eta \sum_{t=1}^{T}\ell(i,y_t)\right)$.

Note: In Bayesian terminology, $(w_{1,1}, w_{2,1}, \cdots, w_{N,1})$ is known as the "prior belief" or "prior distribution".

The high level idea in running the R-EXPWTS algorithm with non uniform weights is to initialize/maintain a prior on all compound actions in $[N]^T$, such that $\forall (i_1, i_2, \cdots, i_T) \notin \mathscr{E}_{(m)}$ the weight on $(i_1, i_2, \cdots, i_T)$ is very small. Therefore the initial weight assignment for any compound action $(i_1, i_2, \cdots, i_T) \in [N]^T$ set its weight as

$$
\begin{aligned}
w_1'(i_1, i_2, \cdots, i_T) &= \frac{1}{N}\left(\frac{\alpha}{N}\right)^{\#(i_1,i_2,\cdots,i_T)}\left(1-\alpha+\frac{\alpha}{N}\right)^{T-1-\#(i_1,i_2,\cdots,i_T)} \\
&= \frac{1}{N}\left(\frac{\alpha}{N}\right)^{\sum_{s=1}^{T}1\{i_{s+1}\neq i_s\}}\left(1-\alpha+\frac{\alpha}{N}\right)^{T-1-\sum_{s=1}^{T}1\{i_{s+1}\neq i_s\}} \\
&= \frac{1}{N}\prod_{s=1}^{T-1}\left(\frac{\alpha}{N}\right)^{1\{i_{s+1}\neq i_s\}}\left(1-\alpha+\frac{\alpha}{N}\right)^{1\{i_{s+1}=i_s\}}
\end{aligned}
$$

where $\alpha \in (0,1)$ is referred as switching property. For convenience, define the "marinalized" weight notation

$$
w_1'(i_1, i_2, \cdots, i_t) = \sum_{(i_{t+1},\cdots,i_T)} w_1'(i_1, i_2, \cdots, i_T)
$$

Lemma 9.5. *The initial marginalized weights satisfy the following recursion: Let us consider an N actions game with losses* $\in [0,1]$ *and initialize the R-EXPWTS algorithm with initial weights* $(w_{1,1}, w_{2,1}, \cdots, w_{N,1}) \in \delta_N$. *Then*

$$
\begin{aligned}
&w_1'(i_1) = \frac{1}{N}, \forall i_1 \in [N] \\
&w_1'(i_1, i_2, \cdots, i_{t+1}) = w_1'(i_1, i_2, \cdots, i_t)\left((1-\alpha)1\{i_{s+1}=i_s\}+\frac{\alpha}{N}\right)
\end{aligned}
$$

Proof: The proof follows by observing that $\forall (i_1, i_2, \cdots, i_T) \in [N]^T$, the initial weights admit the interpretation:

$$
w_1'(i_1, i_2, \cdots, i_T) = \mathbb{P}[\mathbf{X}_1 = i_1, \mathbf{X}_2 = i_2, \cdots, \mathbf{X}_T = i_T],
$$

the joint distribution of the first T states of a discrete time Markov chain with state space $= [N]$, where the initial state X_1 is uniformly distributed over $[N]$. In other words, $w_1'(i_1) = \mathbb{P}[\mathbf{X}_1 = i_1] = \frac{1}{N}, \forall i_1 \in [N]$ and the transition probabilities of the Markov chain are given by $\mathbb{P}[\mathbf{X}_{t+1} = i_{t+1}/\mathbf{X}_t = i_t] = \left((1-\alpha)1\{i_{t+1}=i_t\}+\frac{\alpha}{N}\right)$. □

Bibliography

[1] Gabor Bartok, David Pal, Csaba Szepesvari, and Istvan Szita. Online learning - CMPUT 654 Course Notes. 2011.

[2] Nicolo Cesa-Bianchi and Gabor Lugosi. Prediction, Learning and Games. Cambridge University Press, 2006.