This homework is due at 23:59, May 12, 2019.

Problem 1

Consider the problem of individual binary sequence prediction with the logarithmic loss. In this problem, we will derive a prediction algorithm that competes with any *Markov expert*.

Recall the following protocol. Let $T \in \mathbb{N}$. For every $1 \leq t \leq T$, the following happen in order.

1. LEARNER announces $\gamma_t \in [0,1]$.
2. REALITY announces $\omega_t \in \{0,1\}$.

Define the loss function

$$\lambda(\omega,\gamma) \coloneqq -\omega \log \gamma - (1-\omega)\log(1-\gamma), \quad \forall \omega \in \{0,1\}, \gamma \in [0,1].$$

For any $h : \{0,1\}^* \to [0,1]$, we define the associated regret

$$R_T(h) \coloneqq \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t) - \sum_{t=1}^{T} \lambda(\omega_t, h(\omega_{1:t-1})),$$

where $\omega_{1:t-1}$ denotes the string $\omega_1 \ldots \omega_{t-1}$.

1. (20 points) Let $k \in \mathbb{N}$. Let $\mathscr{H}_k$ be the class of *k-th order stationary Markov experts*. That is, every hypothesis $h \in \mathscr{H}_k$ satisfies

   $$h(\omega_{1:t-1}) = h(\omega_{t-k:t-1}), \quad \forall t \in \mathbb{N}.$$

   We add an arbitrary prefix $\omega_{-k+1} \ldots \omega_0$, so every h is well-defined for all t.

   Consider the following algorithm. For every $t \in \mathbb{N}$ and $y_{1:k} \in \{0,1\}^k$, define

   $$n_0(t; y_{1:k}) \coloneqq \sum_{\tau=1}^{t} \mathbb{1}_{\{\omega_{\tau-k:\tau-1} = y_{1:k}, \omega_\tau = 0\}}, \quad n_1(t; y_{1:k}) \coloneqq \sum_{\tau=1}^{t} \mathbb{1}_{\{\omega_{\tau-k:\tau-1} = y_{1:k}, \omega_\tau = 1\}},$$

   where $\mathbb{1}$ denotes the indicator function. For every $t \in \mathbb{N}$, the algorithm outputs

   $$\gamma_t = f_k(\omega_{1:t-1}) \coloneqq \frac{n_1(t-1; \omega_{t-k:t-1}) + 1}{n_0(t-1; \omega_{t-k:t-1}) + n_1(t-1; \omega_{t-k:t-1}) + 2}.$$

   Show the algorithm satisfies

   $$R_T(h) \leq \sum_{\theta \in \{0,1\}^k} \log\left(n_0(T;\theta) + n_1(T;\theta) + 1\right), \quad \forall h \in \mathscr{H}_k.$$

2. (10 points) **Show the regret bound in the previous problem leads to**

   $$R_T(h) \leq 2^k \log\left(1 + \frac{T}{2^k}\right), \quad \forall h \in \mathscr{H}_k.$$

3. (10 points) The following lemma is due to Leung-Yan-Cheong and Cover.

   Lemma 1 ([1])**.** *Define the* log-star function *as*

   $$\log_2^* x \coloneqq \log_2 x + \log_2 \log_2 x + \cdots + \log_2^{w_2^*(x)} x, \quad \forall x \geq 1,$$

   where $w_2^*(x)$ denotes the largest integer w such that $\log_2^w x \geq 0$, and $\log_2^w$ denotes the w-fold composition of the function $\log_2$. (Notice the log-star function is not the iterated logarithm function in computer science.) Then, it holds that

   $$d \coloneqq \sum_{j \in \mathbb{N}} 2^{-\log_2^* j} < +\infty.$$

Problem 1 continued on the next page...

Use Lemma 1 to show there exists an algorithm that achieves

$$R_T(h) \leq 2^k \log\left(1+\frac{T}{2^k}\right) + (\log 2)\left(\log_2 d + \log_2^* k\right), \quad \forall h \in \mathscr{H}_k, \forall k \in \mathbb{N}.$$

Specify the algorithm.

Problem 2

(20 points) We have introduced some applications of *learning with expert advice* in Lecture 7. Find one more application in *published papers* that does not appear in Lecture 7. **Describe the application, address its importance, show how it can be formulated as learning with expert advice, and give a proper citation.**

Problem 3

In this problem, we will study a learning-with-expert-advice algorithm arguably simpler than the aggregating algorithm.

Let Ω be the outcome space and Γ the prediction space. Let $\lambda : \Omega \times \Gamma \to \mathbb{R}$. Let $T \in \mathbb{N}$. For every $1 \leq t \leq T$, the following happen in order.

1. EXPERT-i announces $\gamma_t(i) \in \Gamma$, $1 \leq i \leq n$.
2. LEARNER announces $\gamma_t \in \Gamma$.
3. REALITY announces $\omega_t \in \Omega$.

Let $(w_1(i))_{1\leq i\leq n}$ be a probability vector in $\mathbb{R}^n$, and define

$$W_1 := \sum_{1\leq i\leq n} w_1(i) = 1.$$

The algorithm we consider announces, for every t,

$$\gamma_t := \sum_{1\leq i\leq n} \frac{w_t(i)\gamma_t(i)}{W_t},$$

and after seeing ω_t, compute $w_{t+1}(i)$ and W_{t+1} as

$$w_{t+1}(i) = w_t(i)\mathrm{e}^{-\eta\lambda(\omega_t,\gamma_t(i))}, \quad W_{t+1} = \sum_{1\leq i\leq n} w_{t+1}(i).$$

for some $\eta > 0$.

We assume that $\lambda(\omega,\cdot)$ is η-*exp-concave* for all $\omega \in \Omega$; that is, the mapping $\gamma \mapsto \mathrm{e}^{-\eta\lambda(\omega,\gamma)}$ is concave for all $\omega \in \Omega$.

1. (10 points) **Compare the algorithm with the aggregating algorithm.**
2. (10 points) Define

$$U_t := \frac{-1}{\eta}\log W_t.$$

Show that

$$\lambda(\omega_t,\gamma_t) \leq U_{t+1} - U_t, \quad \forall 1 \leq t \leq T.$$

Problem 3 continued on the next page...

3. (10 points) **Show that**

$$\sum_{t=1}^{T} \lambda(\omega_t, \gamma_t) \leq \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t(i)) + \frac{1}{\eta} \log \frac{1}{w_1(i)}, \quad \forall 1 \leq i \leq n.$$

4. (10 points) **Show that the algorithm considered in this problem can yield a larger regret bound compared to the aggregating algorithm.**

   HINT: Consider the Brier loss.

References

[1] LEUNG-YAN-CHEONG, S. K., AND COVER, T. M. Some equivalences between Shannon entropy and Kolmogorov complexity. *IEEE Trans. Inf. Theory IT-24*, 3 (1978), 331–338.