ECE 8803: Online Decision Making in Machine Learning

Homework 1

Released: Sep 1 *Due: Sep 14, 11:59pm ET*

Objective. To appreciate at a deeper level what is required for adversarial prediction to work, and learn about connections between seemingly different algorithms for the prediction problem. This homework also aims to familarize students with the new concept of regret.

Problem 1 (Linear regret for binary prediction). 10 points In Lecture 3, we saw that an adversarial sequence could be designed for various algorithms (FTL, Periodic-FTL and pure guessing) to make them highly suboptimal. We discussed this in a somewhat qualitative sense by constructing adversarial sequences that force errors, but mentioned that all of these algorithms suffered linear regret. In this problem, we will show this through a series of steps.

(a) In lecture, we saw that FTL on the sequence

$$X_t = \begin{cases} 1 \text{ if } t \text{ odd} \\ 0 \text{ if } t \text{ even} \end{cases}$$

would make an error on every round. Show that this implies *linear regret*, in the sense that $R_T \geq cT$ for some constant $c > 0$ that does not depend on T.

(b) In lecture, we introduced FTL with the tie-breaking rule of predicting $\widehat{X}_t = 0$ if $\widehat{P}_t = 1/2$. Suppose we instead chose the opposite tie-breaking rule, i.e. $\widehat{X}_t = 1$ if $\widehat{P}_t = 1/2$. Design a sequence $\{X_t\}_{t=1}^T$ for which this variant of FTL will incur linear regret. (Be explicit about the sequence, but feel free to borrow steps from the previous sub-part if needed for the derivation.)

(c) In class, we showed that *any* deterministic algorithm, given by prediction functions $f_{\mathsf{det.}} : \{0,1\}^{t-1} \to \{0,1\}$ for every $t \geq 1$, would incur a total loss of T in the worst case by constructing an explicit adversarial sequence. Show that that sequence will lead to linear regret in the sense defined above.

Hint: Can you upper bound the best loss in hindsight for an arbitrary sequence?

(d) (BONUS - 5 points) Show that pure guessing, i.e. predicting $\widehat{X}_t = 1$ with probability $1/2$ on every round, incurs linear regret in the worst case.

(e) (BONUS - 5 points) Can you design a sequence for which pure guessing incurs *zero* regret?

Problem 2 (Follow-the-Leader on stochastic data). 15 points In Lecture 3, we saw that while FTL is a *very* poor choice of prediction algorithm in the worst case owing to its determinism, it actually does well and is in fact the optimal algorithm for a stochastic sequence, incurring only a constant regret on average! In this problem, we unpack its performance further and show that FTL enjoys this nice guarantee not only in expectation, but also with high probability on the randomness in the sequence.

For the entirety of this problem, we consider the stochastic sequence X_t i.i.d. $\sim$ Bernoulli(p) where $p > 1/2$. All expectations and probabilities will be only with respect to the randomness in the sequence, as FTL is a deterministic algorithm.

(a) Use Hoeffding's inequality and the union bound over time steps $t = t_0, \ldots, T$, to upper bound the probability of the following "bad event": FTL predicts $\widehat{X}_t = 0$ on *any* of the time steps $t = t_0, \ldots, T$.

(b) Consider the complementary "good event" under which FTL predicts $\widehat{X}_t = 1$ *for all* $t = t_0, \ldots, T$. Can you upper-bound the regret incurred by FTL on all sequences satisfying this event for a given t_0?

Hint: Remember what the identity of the best constant predictor in hindsight will be under this event.

(c) Suppose that you want to guarantee a certain regret bound with probability *at least* $1 - \delta$ for a particular $\delta \in (0, 1)$. Select a value of t_0 based on δ, T, p and use the above steps to derive such a regret bound that depends on δ, T, p.

(d) (BONUS - 5 points) Hoeffding's inequality was heavily used in the proofs both of regret in expectation (over the randomness in X_t) and high-probability regret. Suppose that we had instead tried to use Chebyshev's inequality. What is the ensuing bound on *expected regret*? Is it as good as the bound we derived in lecture?

Note: Focus especially on the ensuing dependence on T.

Problem 3 ("Anytime" multiplicative weights). 15 points In Lectures 3 and 4, we learned about the multiplicative weights algorithm (MWA) for a fixed step size η, and showed that it satisfies $\Theta(\sqrt{T})$ regret for the choice $\eta = 1/\sqrt{T}$. This implementation of multiplicative weights requires apriori knowledge of the number of rounds T. It would be great if we could modify the algorithm to run "anytime", by not knowing how long the prediction process would run beforehand. It turns out that a simple modification of the algorithm makes this possible. We will explore this modification in this problem.

(a) (Divide the time interval) Assume that T is divisible by 3, and split the time horizon into 2 parts: $[1, \frac{T}{3}]$, $[\frac{T}{3}+1, T]$. (Note that the length of the two intervals is $\frac{T}{3}$, and $\frac{2T}{3}$ respectively.) Then, we consider applying the update of MWA respectively in each interval in the following way: use $\eta = 1/\sqrt{\frac{T}{3}}$ when at the first interval, and *restart* MWA with $\eta = 1/\sqrt{\frac{2T}{3}}$ when at the second interval. (By restarting, we mean that we start the sequence prediction as though the first realization of the sequence was at time step $\frac{T}{3}+1$.)

Prove that the total regret for this algorithm is still $O(\sqrt{T})$.

Hint: can you apply the bound from lecture to bound the regret of the algorithm on the interval $[1, \frac{T}{3}]$*? What about* $[\frac{T}{3}+1, T]$*?*

(b) (Doubling trick) We now apply the idea above to the case where T is not known beforehand. The main idea is to divide the whole time horizon into different exponentially increasing sub-parts, and apply the MWA at each part. We will assume $T = 2^m - 1$, where $m = \log_2(T+1)$ is an integral. Then, we could divide $[1, T]$ to m intervals given by: $[1,1], [2,3], [4,7], \ldots, [2^k, 2^{k+1}-1], \ldots, [2^{m-1}, 2^m-1]$. For interval number k (which is $[2^{k-1}, 2^k - 1]$), since the length is already known, we can run MWA with learning rate $\eta_k = \frac{1}{\sqrt{2^{k-1}}}$, and restart MWA at interval $k+1$ with the new learning rate η_{k+1}. Clearly, this algorithm does not need to know T in advance.

Extend the analysis of the previous sub-part to show that the total regret for this algorithm is still $O(\sqrt{T})$.

Hint: You may find reviewing properties about the geometric sum $\sum_{k=1}^{m} \alpha^k$*, where* $\alpha < 1$*, useful.*

(c) This approach is commonly called the "doubling trick" in online learning literature. Does it preserve the "multiplicative" nature of the update on weights? Why or why not?

Note: You only need to solve ***one*** *out of Problems 4 or 4$'$. Choose the one that appeals more to you.*

Problem 4 (FTPL = multiplicative weights, seen one way). 10 points In Lecture 5, we introduced the Follow-the-Perturbed Leader (FTPL) algorithm, and proved that it achieves $\Theta(\sqrt{T})$ regret in quite a different approach. In this problem, we will show a surprising connection between this algorithm and the multiplicative weights algorithm. Recall that for binary sequence prediction, the FTPL update is given by

$$\widehat{X}_t = \arg\min_{x \in \{0,1\}} \left[L_{t-1,x} + N_{t,x}\right]$$

where $N_{t,x}$ are iid random variables across time steps t and $x \in \{0,1\}$, drawn according to a distribution parameterized by learning rate η. In class, we looked at the example of the exponential distribution: we will examine a different choice here.

(a) We will consider the case of the *Gumbel* distribution, i.e. $N_{t,x} \sim \mathsf{Gumbel}(\eta)$ where the PDF of the Gumbel distribution parameterized by η is given by $p_{\mathsf{Gumbel}}(x;\eta) = \eta \exp(-\eta x^{-\eta x})$. First, for two random variables X_1, X_2 i.i.d $\sim \mathsf{Gumbel}(\eta)$, evaluate the CDF in the difference between the random variables $N_{t,1} - N_{t,0}$.

(b) Use this fact to explicitly evaluate the expression for $P_t := \mathbb{P}[\widehat{X}_t = 1]$. What does it look like?

Hint: Write P_t in terms of the CDF of $N_{t,1} - N_{t,0}$!

Problem 4' (FTPL = multiplicative weights, seen another way). 10 points In this problem, we will show the connection between FTPL and the multiplicative weights algorithm in a different way.

(a) Let $u_{t,x}$ be a uniform random variable over $[0,1]$, where $x \in \{0,1\}$, and let $s_{t,i} = \ln \frac{1}{u_{t,i}}$. Prove that

$$\widehat{X}_t = \arg\max_{x \in \{0,1\}} \left[\frac{\exp(-\eta L_{t-1,x})}{s_{t,i}}\right]$$

is equivalent to MW.

Hint: First compute $\mathbb{P}[\widehat{X}_t = 1 | s_{t,1}]$, where the probability is over the randomness in the random variable $s_{t,0}$, and then find $\mathbb{P}[\widehat{X}_t = 1]$.

(b) Let $N_{t,x} = \frac{1}{\eta} \ln \ln \frac{1}{u_{t,x}}$. Prove that

$$\widehat{X}_t = \arg\max_{x \in \{0,1\}} \left[\frac{\exp(-\eta L_{t-1,x})}{\ln \frac{1}{u_{t,x}}}\right]$$

is equivalent to

$$\widehat{X}_t = \arg\min_{x \in \{0,1\}} \left[L_{t-1,x} + N_{t,x}\right].$$

Problem 5 (Bonus). 10 points

(a) What do you expect to learn from this class? Please be honest and as detailed as you would like; we will try and adjust our coverage depending on your responses if there is a critical mass of people who would like to learn a particular concept.

(b) Run the Jupyter notebook that was demonstrated in lecture, and experiment with your own choices of various learning rates and types of sequences. Did you observe anything interesting or unexpected? If you work on this sub-part, please provide your modified notebook along with the homework submission.