Fall 2022 *Instructor: Vidya Muthukumar*

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 quantitatively 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. Use the rule that breaks ties in favor of 0 (as we did in Lecture 3).

(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 new 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* L_T^* *for an arbitrary sequence?*

(d) Design a sequence for which pure guessing, i.e. predicting $\widehat{X}_t = 1$ with probability $1/2$ on every round, incurs linear regret. (Recall that you want to evaluate *expected* loss for this algorithm, where the expectation is taken over the randomization in the algorithm.)

(e) Show that pure guessing incurs *zero* regret for the sequence in part (a).

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, it actually does well on a stochastic sequence. We saw a qualitative justification whereby FTL is very unlikely to pick the suboptimal action after a certain number of rounds. In this problem, we unpack its performance more quantitatively and show that FTL achieves low regret *with high probability* on any stochastic 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 not a randomized algorithm.

(a) Fix an integer $t_0 \in \{1, \ldots, T/2 - 1\}$. 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 $t = t_0, \ldots, T$. In other words, can you upper bound the probability that $\widehat{X}_{t_0} = 0$ OR $\widehat{X}_{t_0+1} = 0$ OR ... $\widehat{X}_T = 0$?

(b) Consider any sequence that satisfies the complementary "good event" under which FTL predicts $\widehat{X}_t = 1$ *for all* $t = t_0, \ldots, T$. Show that the regret of FTL for *any* sequence satisfying this event is less than 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 *upper bound* with probability *greater than or equal to* $1 - \delta$ for a particular value of $\delta \in (0, 1)$. In other words, you want to show that $R_T \leq A$ for some upper bound A, with probability $\geq 1 - \delta$. Select a value of t_0 based on δ, T, p and use the above steps to derive such a regret upper bound given by $A(\delta, T, p)$.

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 to pick the step size η in the best possible way. 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 (i.e. between $w_{t+1,x}$ and $w_{t,x}$)? Why or why not?

Problem 4 (Binary sequence prediction with expert advice and the "halving" algorithm.) 10 points In this problem, we investigate the performance of an algorithm that preceded multiplicative weights—the "halving" algorithm. We describe the basic setting, the halving algorithm, and finally the assumption that we will make.

Setting: For this problem, we consider the setting of sequential prediction with expert advice. In other words, we want to predict a binary sequence $X_1, \ldots, X_T$ over time; for example, let this sequence denote whether the weather was sunny (1) or rainy (0) on a given day. To assist us with this prediction, we have experts $1, \ldots, n$, and expert i makes a forecast of $f_{i,t} \in \{0, 1\}$ for round t, e.g. each expert could represent a weather forecaster. At round t, we have access to the experts' forecasts $f_{1,t}, f_{2,t}, \ldots, f_{n,t}$, and we want to somehow *aggregate* these forecasts into our own prediction, which we denote by $\widehat{X}_t$. Once we make our prediction, we observe the true realization X_t, and (as we studied in class) incur the 0-1 loss depending on whether or not we got the prediction right.

Algorithm: First, maintain a set of "active experts" $C_t \subseteq \{1, \ldots, n\}$ at each round. Start with $C_1 = \{1, \ldots, n\}$ for the first round. Then, the halving algorithm performs the following steps at round t:

- *Prediction at round t:* Pick the *majority vote* out of the experts in C_t, e.g. if more experts forecast 1, output $\widehat{X}_t = 1$. If an equal number of experts forecast 1 and 0 (e.g. $|C_t| = 10$, 5 experts forecast 1 and 5 experts forecast 0), then output $\widehat{X}_t = 1$. In other words, the algorithm breaks ties[1] in favor of 1. Finally, if $|C_t| = 1$, i.e. there is only one active expert, output that expert's forecast.
- Observe X_t and evaluate performance. A *mistake* happens when $\widehat{X}_t \neq X_t$.
- *Updating C_{t+1} from C_t:* Remove all experts from C_t who got the prediction wrong, i.e. all experts i for which $f_{i,t} \neq X_t$.
- *If $C_{t+1} = \emptyset$, i.e. $|C_{t+1}| = 0$:* Predict $\widehat{X}_t = 1$ with probability 0.5 and $\widehat{X}_t = 0$ with probability 0.5 on all future rounds. If the algorithm reaches this stage, we evaluate *expected* performance, just as we did in class.

Assumption: For the entire problem, assume that there exists at least one expert i^* which forecasts perfectly on every round, i.e. $f_{i^*,t} = X_t$ for all $t = 1, \ldots, T$.

(a) Consider a round t on which the halving algorithm makes a prediction mistake. Then, recall that all the erroneous experts are dropped from C_t to construct C_{t+1}. Use the halving algorithm's prediction rule to show that $\frac{|C_{t+1}|}{|C_t|} \leq 1/2$. (This constitutes an *upper bound* on the size of C_{t+1} with respect to C_t.)

(b) Use the above step to upper bound the eventual size of the active set of experts, $|C_T|$, in terms of a) the *total number of mistakes* made by the halving algorithm, and b) the total number of experts n.

[1]The answer to this problem would not change if we instead broke ties in favor of 0; this tie-breaking assumption is for ease of exposition.

(c) On the other hand, derive a *lower bound* on $|C_T|$.

Hint: Does the perfect expert get dropped at any stage?

(d) Combine (a) and (b) and show that the number of mistakes made by the halving algorithm is *at most* $\log_2(n)$ (recall that n is the number of experts). Use this to upper bound the regret of the halving algorithm.

Hint: recall the assumption that there exists a perfect expert. What does the regret become in this case?

(e) Now, interpret this instance in the "experts" paradigm that we introduced in class, and consider instead the FTL algorithm on the n experts, i.e. the algorithm that picks the expert that made the fewest mistakes thus far. If there are multiple such leading experts, break ties by predicting the majority vote out of the forecasts provided by the leading experts. Show that the FTL algorithm with this tie-breaking rule is equivalent to the halving algorithm.

Hint: What is the relation between the set of leading experts at round t, and C_t as defined in the halving algorithm?

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.