ECE 8803: Online Decision Making in Machine Learning

Homework 2

Objective. To help build a bridge from prediction to decision-making, and help students interpret online prediction algorithms through the framework of online optimization.

Problem 1 (Multiplicative weights = Follow-the-Regularized-Leader) In this problem, we will show that multiplicative weights can be expressed in the online convex optimization framework, by showing that it is equivalent to a Follow-the-Regularized-Leader problem with a particular type of regularization that encourages randomization. This regularization is called the *entropy* function, and originated in the study of information theory.

(a) For the case of binary prediction, recall that we defined $\widehat{P}_t := ¶[\widehat{X}_t = 1]$. For any $p \in (0, 1)$ define the *binary entropy function*

$$H(p) := -p \log(p) - (1 - p) \log(1 - p).$$

Plot $H(p)$ as a function of p (NOTE: you do not need to consider the cases $p = 0$ and $p = 1$). When is it maximized, and when is it minimized? (You do not need to prove this: the correct answer suffices.)

(b) Recall that the FTL algorithm was given by

$$\widehat{X}_t := \arg\min_{x \in \{0,1\}} L_{t-1,x}, \tag{1}$$

where we defined $L_{t-1,x} := \sum_{s=1}^{t-1} l_{s,x}$ and $l_{s,x} = \mathbb{I}[X_s \neq x]$. Show that this update can be written as the minimum to an optimization problem in the following form:

$$\widehat{P}_t := \arg\min_{p \in [0,1]} [f(p, L_{t-1,0}, L_{t-1,1})]$$

where $f(\cdot)$ is linear in each of $p, L_{t-1,0}$ and $L_{t-1,1}$.

Hint: Note that since FTL is a deterministic algorithm, it induces either $\widehat{P}_t = 0$ (when $\widehat{X}_t = 0$) or $\widehat{P}_t = 1$ (when $\widehat{X}_t = 1$).

(c) Using the function that you defined above, consider the *regularized* update

$$\widehat{P}_t := \arg\min_{p \in [0,1]} f(p, L_{t-1,0}, L_{t-1,1}) - \frac{1}{\eta} H(p).$$

Show that this is equivalent to the multiplicative weights algorithm by explicitly calculating $\widehat{P}_t$.

Hint: Use your observation from part (a) to argue that the optimization problem above is convex. Can you use this to calculate the minimum?

(d) Use your observation in part (a) to argue, in your own words, why a smaller value of η encourages more randomization.

(e) (BONUS – Prediction with expert advice) In the more general setting of prediction with K experts, we can define losses $l_{t,i}$ at time t for each expert $i \in \{1, \ldots, K\}$. Accordingly, we can define a total loss vector $\boldsymbol{L}_t := \begin{bmatrix} L_{t,1} & \ldots & L_{t,K} \end{bmatrix}$ and a probability vector $\boldsymbol{p}_t := \begin{bmatrix} p_{t,1} & \ldots p_{t,K} \end{bmatrix}$. Finally, we can also define a binary entropy function $H(\boldsymbol{p}) := -\sum_{i=1}^{K} -p_i \log(p_i)$.

Repeat parts (a) – (d) to show that the multiplicative weights algorithm for K experts can be written as a *regularized update*, i.e. a solution to the optimization objective $f(\boldsymbol{p}, \boldsymbol{L}_t) - \frac{1}{\eta} H(\boldsymbol{p})$.

Hint: Use your intuition developed from parts (a) – (d) to propose an optimization objective. Then, solve the ***unconstrained*** *minimum of this objective. Do you notice a pattern?*

Answer 1 (Multiplicative weights = Follow-the-Regularized-Leader)

(a) The plot of the entropy function $H(p)$ versus p is given below. Note that you did not need to consider the exact values $p = 0$ and $p = 1$, as this would lead to numerical issues.

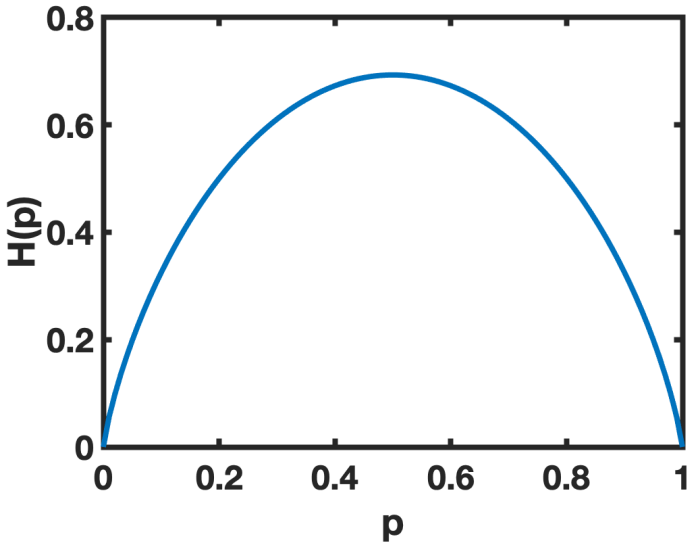

Figure 1: $H(p)$ v.s. p.

It can be observed that the function is maximized when $p = 0.5$, and minimized when $p = 0$ and 1. The question does not ask for this, but you might be interested to know that $\lim_{p \to 0} H(p) = 0$ and $\lim_{p \to 1} H(p) = 1$.

(b) *Any* function of the form $f(p, L_{t-1,0}, L_{t-1,1}) = p(L_{t-1,1} - L_{t-1,0}) + c$ (where c can depend on $L_{t-1,0}$ or $L_{t-1,1}$, but does not depend on p), will give the correct answer. One particularly intuitive choice is given by $f(p, L_{t-1,0}, L_{t-1,1}) = (1-p)L_{t-1,0} + pL_{t-1,1}$. When $L_{t-1,0} < L_{t-1,1}$, we get

$$\widehat{P}_t = \arg \min_{p \in [0,1]} (1-p)L_{t-1,0} + pL_{t-1,1} = 0,$$

and thus $\widehat{X}_t = 0$. On the other hand, when $L_{t-1,0} > L_{t-1,1}$, we have

$$\widehat{P}_t = \arg \min_{p \in [0,1]} (1-p)L_{t-1,0} + pL_{t-1,1} = 1,$$

so $\widehat{X}_t = 1$. Therefore, this algorithm is equivalent to the FTL algorithm.

(c) Note that $H(p)$ is a concave function* with respect to p, and $f(p, L_{t-1,0}, L_{t-1,1})$ is a linear function with respect to p. Thus, the objective function

$$g(p) = f(p, L_{t-1,0}, L_{t-1,1}) - \frac{1}{\eta} H(p)$$

is a convex function of p. Since $\widehat{P}_t$ minimizes $g(p)$, and $g(p)$ is convex in p, we need to have $g'(\widehat{P}_t) = 0$, where $g'(\cdot)$ denotes the derivative of the function $g(\cdot)$. Now, the derivative of $g(\cdot)$ is given by

$$g'(p) = L_{t-1,1} - L_{t-1,0} + \frac{1}{\eta} \log \frac{p}{1-p}.$$

Setting $g'(\widehat{P}_t) = 0$ gives us

$$\frac{1}{\eta} \log \left(\frac{\widehat{P}_t}{1 - \widehat{P}_t} \right) = L_{t-1,0} - L_{t-1,1},$$

which implies that

$$\widehat{P}_t = \frac{\exp(\eta(L_{t-1,0} - L_{t-1,1}))}{1 + \exp(\eta(L_{t-1,0} - L_{t-1,1}))} = \frac{\exp(-\eta L_{t-1,1})}{\exp(-\eta L_{t-1,0}) + \exp(-\eta L_{t-1,1})},$$

which is exactly the multiplicative weights algorithm.

*You do not need to explicitly justify this fact, but here is the rigorous explanation for why $H(p)$ is convex. The second derivative of $H(p)$ can be verified to be:

$$\frac{\partial^2 H(p)}{\partial p^2} = -\frac{1}{p(1-p)} \leq 0$$

for all $p \in (0, 1)$. By the second-order definition of concavity, this means that $H(p)$ is clearly concave in p.

(d) Note that the regularization term $-\frac{1}{\eta} H(p)$ is minimized when $p = 0.5$, and maximized for $p = 0$ or 1. Therefore, this term will force $\widehat{P}_t$ to be close to 0.5. A smaller η means assigning more weights to the regularization term, which leads to more randomization.

(e) We first consider the *unconstrained* optimization problem, given by:

$$\widehat{\mathbf{p}}_t = \arg \min_{\mathbf{p} \in \mathbb{R}_+^K} \mathbf{p}^\top \mathbf{L}_t - \frac{1}{\eta} H(\mathbf{p}),$$

where $H(\mathbf{p}) = \sum_{i=1}^K -p_i \log p_i$ and $\mathbb{R}_+^K$ denotes the non-negative orthant, i.e. all vectors with non-negative entries. We denote the objective function as $g(\mathbf{p})$, and it can be shown that $g(\mathbf{p})$ is convex by using the second-order definition of convexity that we saw in class. In particular, the Hessian can be verified to be:

$$\nabla^2 g(\mathbf{p}) = \begin{bmatrix} \frac{1}{\eta p_1} & & \\ & \ddots & \\ & & \frac{1}{\eta p_K} \end{bmatrix},$$

which is clearly positive semi-definite, since $p_i \geq 0$ for all $i \in \{1, \ldots, K\}$ and we set the learning rate to be $\eta \geq 0$ as well. Thus, because of the convexity of $g(\mathbf{p})$, we have

$$\nabla g(\widehat{\mathbf{p}}_t) = \begin{bmatrix} \frac{1}{\eta}(\log(p_1) + 1) + L_{t-1,1} \\ \vdots \\ \frac{1}{\eta}(\log(p_K) + 1) + L_{t-1,K} \end{bmatrix} = \mathbf{0}.$$

Now, This gives us

$$\widehat{\mathbf{p}}_t = \begin{bmatrix} \exp(-\eta L_{t,1} - 1) \\ \vdots \\ \exp(-\eta L_{t,K} - 1) \end{bmatrix}.$$

We are almost done; however, while $\widehat{\mathbf{p}}_t$ is clearly a non-negative vector, it is *not* a probability vector (i.e. its entries do not sum to 1). Thus, to get our actual optimal solution $\mathbf{p}_t$, we need to *project* $\widehat{\mathbf{p}}_t$ onto the K-dimensional probability simplex, i.e. normalize it so that its entries sum to 1. The natural way to do this is to set:

$$\mathbf{p}_t = \frac{\widehat{\mathbf{p}}_t}{\|\widehat{\mathbf{p}}_t\|_1},$$

which gives us

$$p_i = \frac{\exp(-\eta L_{t,i} - 1)}{\sum_{j=1}^{K} \exp(-\eta L_{t,j} - 1)} = \frac{\exp(-\eta L_{t,i})}{\sum_{j=1}^{K} \exp(-\eta L_{t,j})}$$

for each $i \in \{1, \ldots, K\}$. This is exactly the multiplicative weights algorithm.

Problem 2 (New algorithms through regularization) See attached iPython notebook for details on this problem. Here, we only provide a brief description of the problem and its subparts.

In Problem 1, you showed that multiplicative weights can be written as the solution to the following optimization problem:

$$\hat{P}_t = \arg\min_{p\in[0,1]} \left[f(L_{t-1,1}, L_{t-1,0}, p) - \frac{1}{\eta} H(p) \right],$$

where $H(p)$ was defined as the binary entropy function.

Now, we will explore a different type of regularizer, i.e. instead consider the update

$$\tilde{P}_t = \arg\min_{p\in[0,1]} \left[f(L_{t-1,1}, L_{t-1,0}, p) + \frac{1}{\eta} R(p) \right],$$

where we now define

$$R(p) := \log\left(\frac{1}{p}\right) + \log\left(\frac{1}{1-p}\right).$$

This is often called the "log-barrier" regularizer and was actually first proposed by Nemirovski and Yudin.

(a) Plot $R(p)$ versus p. Where is it minimized? Where is it maximized? Use this plot to argue that minimizing $R(p)$ encourages randomization.

(b) The optimization problem defined for the log-barrier regularizer turns out to have a closed form solution, which is given by:

$$\tilde{P}_t := f_2(D_t) := \frac{2}{\eta D_t + \sqrt{\eta^2 D_t^2 + 4} + 2},$$

where $D_t := L_{t-1,1} - L_{t-1,0}$, i.e. the difference in the cumulative losses.

Express the multiplicative weight update at time step t as a different function $\hat{P}_t = f_1(D_t)$. Then, plot both functions $f_1(\cdot)$ and $f_2(\cdot)$ as a function of d ranging between -10 and 10 for the case $\eta = 1$. Which function decreases faster?

(c) Implement the multiplicative weights update and the log-barrier update here as a function of the losses $L_{t-1,0}, L_{t-1,1}$. In the starter code provided below, the variable "losses" is a numpy array with 2 entries given by $[L_{t-1,0}, L_{t-1,1}]$. Please make the output of your algorithm a numpy array "prob" with 2 entries, given by $[P_{t,0}, P_{t,1}]$.

You are free to refer to (and borrow from) the code provided in the resources "MultiplicativeWeights.ipynb" (note that this was defined for rewards, so will require slight changes).

Hint for log-barrier: recall that we defined $D_t = L_{t-1,1} - L_{t-1,0}$.

(d) Now, we compare the performance of both algorithms on the two running examples that we have been considering in class. We set $T = 1000$, and consider two types of sequences:

a) the case where $X_t = 1$ for all $t = 1, \ldots, T$. b) the case of the 1-periodic sequence.

For both algorithms, we will set $\eta = 1/\sqrt{T}$.

Plot the total *loss* of each algorithm (what we defined as H_t in class) versus the number of time steps t on sequences (a) and (b). Please make separate figures for the evaluation of (a) and (b).

You are welcome to directly use the starter code provided below to evaluate the performance of an arbitrary algorithm (parameterized by eta) on an arbitrary binary sequence. This starter code will return a "total loss vector" given by $\begin{bmatrix} H_1 & \dots H_T \end{bmatrix}$ for plotting convenience.

(e) Report the superior algorithm for sequence (a) and sequence (b) respectively. Based on this report, which algorithm do you think tends to randomize more? (You are also welcome to use the answer to part (b) as a hint.)

(f) (BONUS - theory) Prove that the log-barrier regularizer leads to the explicit update given in part (b).

Hint: the quadratic formula may be useful.

Answer 2 (part f, all other answers in attached Jupyter notebook) The objective function is given by

$$\begin{aligned} g(p) &= (1-p)L_{t-1,0} + pL_{t-1,1} + \frac{1}{\eta}[-\log(p) - \log(1-p)] \\ &= pD_t + L_{t-1,0} + \frac{1}{\eta}[-\log(p) - \log(1-p)], \end{aligned}$$

First, we note that the objective function $g(p)$ is convex in p. This is because the function $-\log(p) - \log(1-p)$ can be verified to be convex (because its second derivative is equal to $\frac{1}{p^2} + \frac{1}{(1-p)^2}$, which is clearly non-negative), and the sum of a convex and a linear function in p is convex.

Thus, the minimum of $g(p)$ is achieved at $g'(p) = 0$. We have

$$g'(p) = D_t + \frac{1}{\eta}\left(-\frac{1}{p} + \frac{1}{1-p}\right) = D_t + \frac{2p-1}{-\eta p^2 + \eta p};$$

setting this to be equal to 0 gives us

$$D_t = \frac{1-2p}{-\eta p^2 + \eta p},$$

which implies that

$$-D_t \eta p^2 + (D_t \eta + 2)p - 1 = 0.$$

Thus, the optimal value $\widehat{P}_t$ is given by:

$$\begin{aligned} \widehat{P}_t &= \frac{-(D_t\eta + 2) + \sqrt{(D_t\eta+2)^2 - 4D_t\eta}}{-2D_t\eta} \\ &= \frac{(D_t\eta + 2) - \sqrt{D_t^2\eta^2 + 4}}{2D_t\eta} \\ &= \frac{D_t^2\eta^2 + 4D_t\eta + 4 - D_t^2\eta^2 - 4}{2D_t\eta[(D_t\eta+2) + \sqrt{D_t^2\eta^2+4}]} \\ &= \frac{2}{\eta D_t + \sqrt{\eta^2 D_t^2 + 4} + 2}. \end{aligned}$$

Above, in the first step, we used the quadratic formula — note that the alternative solution $\frac{-(D_t\eta+2)-\sqrt{(D_t\eta+2)^2-4D_t\eta}}{-2D_t\eta}$ would not lie in $[0,1]$, so is not valid. In the third step, we multiplied numerator and denominator by $(D_t\eta+2)+\sqrt{D_t^2\eta^2+4})$ and used the basic identity $(a+b)(a-b) = a^2-b^2$.

Problem 3 (Lower bound for online linear optimization (OLO)). 15 points We have discussed several algorithms in class that achieve the optimal regret guarantee $R_T = \mathcal{O}(\sqrt{T})$. We have also mentioned that this guarantee is optimal (i.e. there exists an instance on which *any* algorithm would incur at least $\sqrt{T}$ regret). In the setting of binary prediction, or decision-making using expert advice, this instance is *randomly constructed* and somewhat complicated to describe. However, in the setting of online linear optimization, it turns out that we can create a remarkably simple adversary that forces *any* OLO algorithm to incur exactly $\sqrt{T}$ regret. In this problem, we will show precisely this. All norms $\|\cdot\|$ are, by default, the Euclidean (ℓ_2) norm.

(a) Let $\{\boldsymbol{\ell}_t\}_{t=1}^T$ be a set of vectors in $\mathbb{R}^d$. Assume that $\|\boldsymbol{\ell}_i\| = 1$ for all $t = 1, \ldots, T$, and furthermore, $\boldsymbol{\ell}_t^\top (\sum_{s=1}^{t-1} \boldsymbol{\ell}_s) = 0$ for all $t > 1$. Prove that $\|\sum_{t=1}^T \boldsymbol{\ell}_t\| = \sqrt{T}$.

Hint: You may find it useful to start from the fact that

$$\left\|\sum_{t=1}^{T} \boldsymbol{\ell}_t\right\|^2 = \left\|\sum_{t=1}^{T-1} \boldsymbol{\ell}_t + \boldsymbol{\ell}_T\right\|^2 = \left\|\sum_{t=1}^{T-1} \boldsymbol{\ell}_t\right\|^2 + \|\boldsymbol{\ell}_T\|^2 + 2\boldsymbol{\ell}_T^\top \left(\sum_{t=1}^{T-1} \boldsymbol{\ell}_t\right)$$

where the above uses the algebraic identity on $\|\boldsymbol{a} + \boldsymbol{b}\|^2$ for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. After simplifying the above, apply the same procedure recursively, i.e. to the term $\left\|\sum_{t=1}^{T-1} \boldsymbol{\ell}_t\right\|^2$.

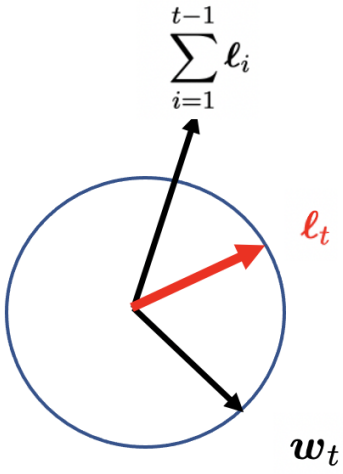

Figure 2: Example when $d = 3$.

(b) Consider the case where $d \geq 3$, and let the decision set $\mathcal{B}$ be a ball of radius 1 and centered at the origin. In other words, $\mathcal{B} := \{\boldsymbol{w} : \|\boldsymbol{w}\| \leq 1\}$. Prove that for *any* set of loss vectors $\{\boldsymbol{\ell}_t\}_{t=1}^T$, the optimal decision in hindsight is given by:

$$\boldsymbol{w}_* = -\frac{\sum_{t=1}^T \boldsymbol{\ell}_t}{\|\sum_{t=1}^T \boldsymbol{\ell}_t\|}.$$

(c) Now, we define our the adversary's policy is defined as follows: When $t = 1$, after observing the learner's decision $\boldsymbol{w}_1$, the adversary chooses a vector $\boldsymbol{\ell}_1$ such that $\|\boldsymbol{\ell}_1\| = 1$ and $\boldsymbol{\ell}_1^\top \boldsymbol{w}_1 = 0$. For

round $t \geq 2$, the adversary will choose a $\boldsymbol{\ell}_t$ such that $\|\boldsymbol{\ell}_t\| = 1$, $\boldsymbol{\ell}_t^\top \boldsymbol{w}_t = 0$ and $\boldsymbol{\ell}_t^\top (\sum_{i=1}^{t-1} \boldsymbol{\ell}_i) = 0$. Note that such a $\boldsymbol{\ell}_t$ can always be found for $d \geq 3$ (as $\boldsymbol{\ell}_t$ is the normal vector of the plane spanned by $\boldsymbol{w}_t$ and $\sum_{i=1}^{t-1} \boldsymbol{\ell}_i$). Figure 1 shows an example for the 3-dimensional case.

Prove that the regret of *any* OLO algorithm *is equal to* $\sqrt{T}$ under this policy.

(d) (BONUS – 10 points) For $d = 2$, can we still design a policy for the adversary to ensure the $\Omega(\sqrt{T})$ lower bound?

Hint: In fact, $\boldsymbol{\ell}_t$ does not need to be exactly orthogonal to $\boldsymbol{w}_t$ and $\sum_{i=1}^{t-1} \boldsymbol{\ell}_i$ for the proof approach to work.

Answer 3 (Lower bound for OLO)

(a) As per the hint, we solve this problem in steps by first showing that

$$\left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|^2 = \left\| \sum_{i=1}^{T-1} \boldsymbol{\ell}_t \right\|^2 + 1.$$

Our first step is to apply the algebraic identity $||\mathbf{a} + \mathbf{b}||^2 = ||\mathbf{a}||^2 + ||\mathbf{b}||^2 + 2\mathbf{a}^\top \mathbf{b}$. We apply this identity with $\mathbf{a} := \sum_{t=1}^{T-1} \boldsymbol{\ell}_t$ and $\mathbf{b} := \boldsymbol{\ell}_T$ to get:

$$\left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|^2 = \left\| \sum_{i=1}^{T-1} \boldsymbol{\ell}_t + \boldsymbol{\ell}_T \right\|^2 = \left\| \sum_{i=1}^{T-1} \boldsymbol{\ell}_t \right\|^2 + \|\boldsymbol{\ell}_T\|^2 + 2\boldsymbol{\ell}_T^\top \left(\sum_{i=1}^{T-1} \boldsymbol{\ell}_t \right) = \left\| \sum_{i=1}^{T-1} \boldsymbol{\ell}_t \right\|^2 + 1,$$

where in the last step we use the fact that our vectors are defined such that $\boldsymbol{\ell}_T^\top \left(\sum_{i=1}^{T-1} \boldsymbol{\ell}_t \right) = 0$. Thus, we have proved the required statement. We can then apply this argument recursively to get

$$\left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|^2 = \left\| \sum_{i=1}^{T-2} \boldsymbol{\ell}_t \right\|^2 + 2,$$

and so on and so forth until we get $\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \|^2 = T$. This then gives us $\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \| = \sqrt{T}$.

(b) Either the geometric or algebraic solution is acceptable for this sub-part.

(Geometric solution) Since $\mathcal{B}$ is a ball of radius 1 and centered at the origin, we have

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w} \in \mathcal{B}} (\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w}$$

should be the vector on the surface of $\mathcal{B}$, and its direction is opposite to the vector $\sum_{t=1}^{T} \boldsymbol{\ell}_t$. Thus, we have

$$\boldsymbol{w}^* = -\frac{\sum_{t=1}^{T} \boldsymbol{\ell}_t}{\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \|}.$$

(Algebraic solution) Firstly, note that $\|\boldsymbol{w}\| \leq 1$ is equivalent to $\|\boldsymbol{w}\|^2 \leq 1$. Thus, essentially, $\boldsymbol{w}^*$ can be found by solving the following constrained optimization problem:

$$\min_{\boldsymbol{w} \in \{\boldsymbol{w}: \|\boldsymbol{w}\|^2 \leq 1\}} (\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w},$$

which is equivalent to

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \max_{\lambda \geq 0} L(\boldsymbol{w}, \lambda) = (\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w} + \lambda(\|\boldsymbol{w}\|^2 - 1).$$

Let λ^* and $\boldsymbol{w}^*$ be the optimal solution of the new problem above. Based on the KKT condition, we know the gradient with respect to $\boldsymbol{w}$

$$\nabla L(\boldsymbol{w}^*, \lambda^*) = \sum_{t=1}^{T} \boldsymbol{\ell}_t + 2\lambda^* \boldsymbol{w}^* = 0,$$

which implies

$$\boldsymbol{w}^* = -\frac{1}{2\lambda^*} \sum_{t=1}^{T} \boldsymbol{\ell}_t.$$

On the other hand, KKT condition also leads to

$$\lambda^*(\|\boldsymbol{w}^*\|^2 - 1) = 0.$$

Combining the two equalities above, we have

$$\lambda^* = \frac{1}{2} \left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|,$$

and thus

$$\boldsymbol{w}^* = -\frac{\sum_{t=1}^{T} \boldsymbol{\ell}_t}{\left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|}.$$

(c) We recall that the regret is defined as

$$R_T = H_T - L_T^*, \text{ where}$$
$$H_T := \sum_{t=1}^{T} \boldsymbol{\ell}_t^\top \boldsymbol{w}_t \text{ and}$$
$$L_T^* := \min_{\boldsymbol{w} \in \mathcal{B}} \sum_{t=1}^{T} \boldsymbol{\ell}_t^\top \boldsymbol{w}.$$

Since the adversary's construction gives us $\boldsymbol{\ell}_t^\top \boldsymbol{w}_t = 0$ for all values of t, we directly get $H_T = 0$. For the second term L_T^*, we have

$$-L_T^* = -(\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w}^* = \left(\sum_{t=1}^{T} \boldsymbol{\ell}_t \right)^\top \frac{\sum_{t=1}^{T} \boldsymbol{\ell}_t}{\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \|} = \left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\| = \sqrt{T},$$

where the last step is based on the answer to part (a). Summing these two terms completes the proof.

(d) The adversary's policy is as follows: When $t = 1$, after observing the learner's decision $\boldsymbol{w}_1$, the adversary chooses a vector $\boldsymbol{\ell}_1$ such that $\|\boldsymbol{\ell}_1\| = 1$ and $\boldsymbol{\ell}_1^\top \boldsymbol{w}_1 = 0$. For $t \geq 2$, the adversary will choose a $\boldsymbol{\ell}_t$ such that $\|\boldsymbol{\ell}_t\| = 1$, $\boldsymbol{\ell}_t^\top \boldsymbol{w}_t \geq 0$ and $\boldsymbol{\ell}_t^\top (\sum_{i=1}^{t-1} \boldsymbol{\ell}_i) \geq 0$.

The key idea that we are using is that we only need *inequality* constraints above, not equality constraints. We note that these inequalities can always be satisfied in the 2 dimensional case, as geometrically pictured below:

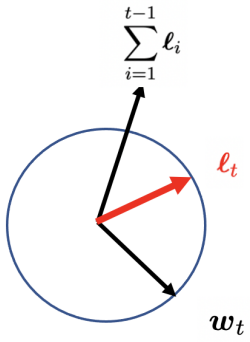

Figure 3: Example for $d = 2$.

For the first term of regret H_T, we have

$$H_T = \sum_{t=1}^{T} \boldsymbol{\ell}_t^\top \boldsymbol{w}_t \geq 0.$$

For the second term of regret L_T^*, we note that

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w} \in \mathcal{B}} (\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w}$$

is still

$$\boldsymbol{w}^* = -\frac{\sum_{t=1}^{T} \boldsymbol{\ell}_t}{\|\sum_{t=1}^{T} \boldsymbol{\ell}_t\|},$$

so we still have

$$-L_T^* = -(\sum_{t=1}^{T} \boldsymbol{\ell}_t)^\top \boldsymbol{w}^* = \left(\sum_{t=1}^{T} \boldsymbol{\ell}_t\right)^\top \frac{\sum_{t=1}^{T} \boldsymbol{\ell}_t}{\|\sum_{t=1}^{T} \boldsymbol{\ell}_t\|} = \left\|\sum_{t=1}^{T} \boldsymbol{\ell}_t\right\|,$$

and

$$\left\|\sum_{t=1}^{T} \boldsymbol{\ell}_t\right\|^2 = \left\|\sum_{i=1}^{T-1} \boldsymbol{\ell}_t + \boldsymbol{\ell}_T\right\|^2 = \left\|\sum_{i=1}^{T-1} \boldsymbol{\ell}_t\right\|^2 + \|\boldsymbol{\ell}_T\|^2 + 2\boldsymbol{\ell}_T^\top \left(\sum_{i=1}^{T-1} \boldsymbol{\ell}_t\right) \geq \left\|\sum_{i=1}^{T-1} \boldsymbol{\ell}_t\right\|^2 + 1,$$

which implies that

$$\left\| \sum_{t=1}^{T} \boldsymbol{\ell}_t \right\|^2 \geq T,$$

so the second term is lower bounded by $\sqrt{T}$. Combining the the two parts, we conclude that the regret is lower-bounded by $\sqrt{T}$.

Problem 4 (Online gradient descent with a time-varying, data-dependent learning rate) 30 points In this problem, we consider solving online convex optimization by a slightly different algorithm, i.e., online gradient descent (OGD). The basic procedure of OGD is as follows: We begin with $\boldsymbol{w}_1 = 0$. For round $t = 1, \ldots, T$, OGD firstly observes the gradient of $\ell_t(\cdot)$ at $\boldsymbol{w}_t$, i.e., $\nabla \ell_t(\boldsymbol{w}_t)$, then preforms a descent step towards the gradient direction with learning rate $\eta_t > 0$,

$$\boldsymbol{z}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla \ell_t(\boldsymbol{w}_t),$$

and next projects the result into the decision set to get $\boldsymbol{w}_{t+1}$:

$$\boldsymbol{w}_{t+1} = \Pi_{\mathcal{B}}[\boldsymbol{z}_{t+1}],$$

where

$$\Pi_{\mathcal{B}}[\boldsymbol{z}] = \arg\min_{\boldsymbol{w} \in \mathcal{B}} \|\boldsymbol{w} - \boldsymbol{z}\|_2^2$$

denotes projecting $\boldsymbol{z}$ onto a convex set $\mathcal{B}$.

This algorithm is related to FTRL, but slightly different. In FTRL, we ran the update from the *unconstrained* decision vector $\boldsymbol{z}_t$ as:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \eta \nabla \ell_t(\boldsymbol{w}_t) \tag{2}$$

and then projected $\boldsymbol{z}_{t+1}$ onto the convex set $\mathcal{B}$. We also considered a fixed learning rate η. However, we will now show that these algorithms have similar performance, and OGD allows us to consider a time-varying, data-dependent learning rate. You will find it useful to review the notes of Lectures 8 and 9 for answering this question.

(a) If we set the learning rate of OGD as $\eta_1 = \eta_2 = \cdots = \eta > 0$, and also ignore the projection steps in OGD and FTRL (review Equation 9.6 for the latter), then what is the relationship between the two algorithms?

(b) Let $\boldsymbol{w}^*$ be the optimal decision in hindsight, defined as $\boldsymbol{w}_* = \arg\min_{\boldsymbol{w} \in \mathcal{B}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t)$. Prove that for every value of $t \geq 1$, we have:

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2 \leq \|\boldsymbol{w}_t - \boldsymbol{w}_*\|_2^2 + \eta_t^2 \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 - 2\eta_t (\boldsymbol{w}_t - \boldsymbol{w}_*)^\top \nabla \ell_t(\boldsymbol{w}_t). \tag{3}$$

Hint: (1) First prove that the right hand side of the above equation is ***equal to*** $\|\boldsymbol{z}_{t+1} - \boldsymbol{w}^*\|_2^2$ *using an appropriate quadratic identity on* $\|\boldsymbol{a} - \boldsymbol{b}\|_2^2$*. Then review Lecture 8 (Figure 8.1, and the surrounding discussion) to recall what projection does to the distance.*

(c) Based on (3), prove that $\forall t \geq 1$,

$$\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}_*) \leq \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\eta_t} + \frac{\eta_t}{2}\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2.$$

Hint: Review Section 9.2.1, and make use of the first-order condition for convex functions to start the proof. More specifically, if $f(\cdot) : \mathcal{B} \mapsto \mathbb{R}$ is convex, then for any $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{B}$, we have

$$f(\boldsymbol{z}_1) \geq f(\boldsymbol{z}_2) + (\boldsymbol{z}_1 - \boldsymbol{z}_2)^\top \nabla f(\boldsymbol{z}_2).$$

Then, combine your observation with the inequality that was derived in part (b).

(d) Based on (c), prove that the regret of OGD can be upper bounded by

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \leq \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\eta_1} + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2.$$

Hint: Sum up the upper bounds derived in part (c). Then, relate the quantities $\sum_{t=1}^{T} \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{\eta_t}$ and $\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2$. You may find it helpful to do this for the case $T = 3$ first, to get intuition.

(e) Let $\eta_t = \frac{D}{G\sqrt{t}}$, where G is an upper bound on $\|\nabla \ell_t(\boldsymbol{w})\|_2$ for all values of t. Prove that the regret bound of OGD is $O(DG\sqrt{T})$,
Hint:
(1) note that $\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \geq 0$, so $\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2 \leq 4D^2\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)$.
(2)We have $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$.

(f) (BONUS – 5 points) Let $\eta_t = \frac{D}{\sqrt{\sum_{i=1}^{t}\|\nabla \ell_i(\boldsymbol{w}_i)\|^2}}$. Prove that the regret bound of OGD is on the order of $O(D\sqrt{\sum_{t=1}^{T}\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2})$. Is this bound better or worse than the bound that you derived in part (e)?
Hint: For nonnegative $a_1, \ldots, a_t$, we have $\sum_{i=1}^{t}\frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \leq 2\sqrt{\sum_{i=1}^{t} a_i}$. Also note that the learning rate, as set, is decreasing with t.

Answer 4 (Online gradient descent with a time-variant learning rate) 30 points

(a) This sub-part states that we can ignore the projection step in both algorithms. Then, considering the learning rate of OGD as $\eta_t = \frac{\eta}{2}$, we get

$$\begin{aligned} \boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta}{2}\nabla\ell_t(\boldsymbol{w}_t) &= \boldsymbol{w}_{t-1} - \frac{\eta}{2}(\nabla\ell_t(\boldsymbol{w}_t) + \nabla\ell_{t-1}(\boldsymbol{w}_{t-1})) \\ &= \dots \\ &= \boldsymbol{w}_1 - \frac{\eta}{2}\sum_{i=1}^{t}\nabla\ell_i(\boldsymbol{w}_i) = -\frac{\eta}{2}\sum_{i=1}^{t}\nabla\ell_i(\boldsymbol{w}_i), \end{aligned} \tag{4}$$

where the last equality is because we initialize at $\boldsymbol{w}_1 = 0$. On the other hand, in FTRL, we have

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{t}\widehat{\ell}_i(\boldsymbol{w}) + \frac{1}{\eta}\|\boldsymbol{w}\|_2^2,$$

where

$$\widehat{\ell}_t(\boldsymbol{w}) = \ell_t(\boldsymbol{w}_t) + (\boldsymbol{w} - \boldsymbol{w}_t)^\top\nabla\ell_t(\boldsymbol{w}_t),$$

which is a linear function, and thus convex. Based on the optimality condition, we have

$$\sum_{i=1}^{t}\nabla\widehat{\ell}_i(\boldsymbol{w}_{t+1}) + \frac{2}{\eta}\boldsymbol{w}_{t+1} = 0,$$

i.e.,

$$\sum_{i=1}^{t}\nabla\ell_i(\boldsymbol{w}_i) + \frac{2}{\eta}\boldsymbol{w}_{t+1} = 0,$$

which also implies that

$$\boldsymbol{w}_{t+1} = -\frac{\eta}{2}\sum_{i=1}^{t}\nabla\ell_i(\boldsymbol{w}_i).$$

Therefore, the two algorithms are equivalent in this case.

(b) We have

$$\begin{aligned} \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2 &\le \|\boldsymbol{z}_{t+1} - \boldsymbol{w}^*\|_2^2 \\ &= \|\boldsymbol{w}_t - \eta_t\nabla\ell_t(\boldsymbol{w}_t) - \boldsymbol{w}^*\|_2^2 \\ &= \|\boldsymbol{w}_t - \boldsymbol{w}^* - \eta_t\nabla\ell_t(\boldsymbol{w}_t)\|_2^2 \\ &= \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - 2\eta_t(\boldsymbol{w}_t - \boldsymbol{w}^*)^\top\nabla\ell_t(\boldsymbol{w}_t) + \eta_t^2\|\nabla\ell_t(\boldsymbol{w}_t)\|_2^2, \end{aligned}$$

where the first inequality applies because projection decrease the distance.

(c) The first-order definition of convexity gives us

$$\begin{aligned} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) &\le (\boldsymbol{w}_t - \boldsymbol{w}^*)^\top\nabla\ell_t(\boldsymbol{w}_t) \\ &\le \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\eta_t} + \frac{\eta_t}{2}\|\nabla\ell_t(\boldsymbol{w}_t)\|_2^2, \end{aligned} \tag{5}$$

where the second inequality follows from rearranging the terms in the solution of part (b).

(d) Based on part (c), we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \\
&\leq \sum_{t=1}^{T} \frac{\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_{t+1} - \boldsymbol{w}^*\|_2^2}{2\eta_t} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 \\
&= \frac{1}{2} \sum_{t=2}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 + \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\eta_1} - \frac{\|\boldsymbol{w}_T - \boldsymbol{w}^*\|_2^2}{2\eta_{T-1}} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 \\
&\leq \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\eta_1} + \frac{1}{2} \sum_{t=2}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2.
\end{aligned}
$$

Note that we dropped the negative term $-\frac{\|\boldsymbol{w}_{T+1} - \boldsymbol{w}^*\|_2^2}{2\eta_T}$ at the last step to state the result in a more clean form.

(e) We fix $\eta_t = \frac{\alpha}{\sqrt{t}}$ and write the answer in terms of α. Based on part (d), we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \\
&\leq \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\eta_1} + \frac{1}{2} \sum_{t=2}^{T} (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) \|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 \\
&\leq \frac{4D^2}{2\eta_1} + \frac{4D^2}{2} (\frac{1}{\eta_T} - \frac{1}{\eta_1}) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t \\
&= \frac{2D^2}{\alpha} + \frac{2D^2(\sqrt{T} - 1)}{\alpha} + \frac{G^2 \alpha}{2} (2\sqrt{T} - 1) \\
&\leq \frac{2D^2\sqrt{T}}{\alpha} + \alpha G^2 \sqrt{T}.
\end{aligned}
$$

Note that we drop the negative term $-\frac{G^2\alpha}{2}$ at the last step to make the bound more clear. Next, we fix $\alpha = \frac{D}{G}$. Then, we get

$$R(T) \leq 3DG\sqrt{T}.$$

(f) Based on part (d), we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{w}^*) \\
&\leq \frac{\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2}{2\eta_1} + \frac{1}{2}\sum_{t=2}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})\|\boldsymbol{w}_t - \boldsymbol{w}^*\|_2^2 + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2 \\
&\leq \frac{4D^2}{2\eta_1} + \frac{4D^2}{2}(\frac{1}{\eta_T} - \frac{1}{\eta_1}) + \frac{1}{2}\sum_{t=1}^{T}\frac{\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2}{\sqrt{\sum_{i=1}^{t}\|\nabla \ell_i(\boldsymbol{w}_i)\|_2^2}} \\
&\leq 3D\sqrt{\sum_{t=1}^{T}\|\nabla \ell_t(\boldsymbol{w}_t)\|_2^2}.
\end{aligned}
$$