ONLINE LEARNING

SUT

&

2023

Contents

Chapter 1

Introduction to Online Learning

Online learning is a significant category of machine learning algorithms that involves a learner trying to tackle tasks such as online prediction or decision-making. This is achieved by learning a model or hypothesis from a sequential stream of data instances, one at a time.

During each round, the learner is presented with a question and is expected to provide a corresponding answer to that question. For example, a learner might be assumed to predict a number while an adversary chooses one. Once the learner has made a prediction, they receive the correct answer to the question. The quality of the learner's answer is then evaluated using a loss function, which quantifies the difference between the predicted answer and the correct one. The learner's ultimate objective is to minimize the cumulative loss experienced throughout its execution. In order to accomplish this, the learner may adjust its hypothesis or model after each round in order to improve its accuracy in subsequent rounds. In the exact word, assume a frequent T-round game in which:

- An adversary choose a real number in $y_t \in [0, 1]$ and he kept in secret
- A learner predicts $x_t \in [0, 1]$.
- Let loss function be $\ell(x_t) = \ell_t = (x_t - y_t)^2$ as a typical loss function.
- The cumulative loss is $\sum_{t=1}^{T}(x_t - y_t)^2$.

It appears that the assumptions alone are insufficient to propose an algorithm that meets the criteria for acceptability. Now we review the problem under different circumstances:

First suppose that y_i's are i.i.d random variables from a probability distribution $F(\mu_F, \sigma_F)$. Could we play in a way that minimizes our penalty? To answer this question, assume that F is a

known distribution. Thus

$$\begin{aligned}\mathbb{E}\big[\ell_t(x)\big] &= \mathbb{E}\big[(x-y_t)^2\big] \\ &= \mathbb{E}\big[(x-\mu_F)^2\big] + \mathbb{E}\big[(y_t-\mu_F)^2\big] - 2\mathbb{E}\big[x-\mu_F\big]\mathbb{E}\big[y_t-\mu_F\big] \\ &= (x-\mu_F)^2 + \sigma_F^2.\end{aligned}$$

So, in order to minimize the total loss, let $x_t = \mu_F$ for $1 \le t \le T$, consequently, the total error is equal to $T\sigma_F^2$.

Now, assume that the distribution F is unknown. We want to prove that by a simple algorithm, we can have a total loss of $O(T\sigma_F^2)$ much like the previous section. If we had known the actual mean of the distribution, we would have put all x_t's equal to that. So it is only logical to put x_t equal to an estimate of the mean. We will do this by putting

$$x_i = \frac{1}{i-1}\sum_{t=1}^{i-1} y_t$$

From previous calculations, we have $\mathbb{E}\big[\ell_t(x)\big] = \mathbb{E}[(x-\mu_F)^2] + \sigma_F^2$. And we know that if $x_1, ..., x_n$ are samples from a distribution $Var(\frac{1}{n-1}\sum_{i=1}^{n-1} y_i) = \frac{\sigma_F^2}{n-1}$. This means the expected of the total loss will be equal to $\sum_{i=1}^{T-1}\frac{\sigma_F^2}{i} + T\sigma_F^2 \approx \ln T \times \sigma_F^2 + T\sigma_F^2 = O(T\sigma_F^2)$ So in this case by a simple algorithm, we achieved a total loss of the same order as before. This means that having the exact distribution is not that important in minimizing the total loss function.

Now, assume that y_i's are not necessarily taken from a distribution. Here we find out that assumptions are not enough to measure the efficiency of the proposed algorithms so, we define another parameter for measuring the algorithm's effectiveness which we will call "Regret". It is denoted by

$$\text{Regret}_T = R_T = \sum_{t=1}^{T}(x_t-y_t)^2 - \min_{x\in[0,1]}\sum_{t=1}^{T}(x-y_t)^2$$

In which x_t's are our algorithm's output and y_t's are the actual numbers chosen by the adversary. In other words, R_T is the difference of our loss if we had chosen the best "constant" algorithm and our actual algorithm. The logic behind this comes from the fact that had we known the actual distribution, we would have chosen a constant guess(which is the mean of the distribution) in order to minimize the expected Loss function. Claim: If we have no condition for y_i's(i.e. we don't know if they are i.i.d. or even taken from a distribution) we can still choose x_i's in a way that R_T grows sublinearly with regards to T.

Proof: we know that we should choose x_i based on $y_1, ..., y_{i-1}$ and if we have no conditions on y_j's,

it is only logical to choose x_i to be the mean of all the known y_j's that is

$$x_i = \frac{1}{i-1}\sum_{t=1}^{i-1} y_t$$

This strategy is also referred to as "Follow the Leader" or "FTL". But one could argue that this only could have been a good move when we knew that y_j's are i.i.d and taken from a distribution because this would have been an estimator for the mean of the distribution. Against all odds, we see that this works here as well!

Proof: Denote x_t^* by a minimizer of the $\sum_{i=1}^{t} \ell_i(x)$. We claim that inequality

$$\sum_{i=1}^{T} \ell_i(x_i^*) \leq \sum_{i=1}^{T} \ell_i(x_T^*)$$

holds for every $T \in \mathbf{N}$.

Proof of the claim: we prove this by induction on T. It's easy to see that the base case inequality holds.($\ell_1(x_1^*) \leq \ell_1(x_1^*)$) Now observe that we have: $\sum_{i=1}^{T-1} \ell_i(x_i^*) \leq \sum_{i=1}^{T} \ell_i(x_{T-1}^*) \leq \sum_{i=1}^{T-1} \ell_i(x_T^*)$ The first inequality holds by induction hypothesis and the second one holds because by the extremal choice of x_{T-1}^* we have $\sum_{i=1}^{T} \ell_i(x_{T-1}^*) \leq \sum_{i=1}^{T-1} \ell_i(x)$ holds for every x which means it would be also true for $x = x_T^*$. So the proof of the claim is complete.(by adding $\ell_T(x_T)$ to both sides, we obtain the inequality for $i = T$ holds)

As previously we mentioned let us run the algorithm in which we allocate the mean of $y_1, y_2, .., y_{t-1}$ to x_t.Then we have:

$$Regret_T = \sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 \leq 4 + 4\ln T$$

Now note that $x_i = x_{i-1}^*$.

$$\begin{aligned}
(x_t - y_t)^2 - (x_t^* - y_t)^2 &= (x_{t-1}^* - y_t)^2 - (x_t^* - y_t)^2 \\
&= (x_{t-1}^* - x_t^*)(x_{t-1}^* + x_t^* - 2y_t) \\
&\leq |x_{t-1}^* - x_t^*||x_{t-1}^* + x_t^* - 2y_t| \\
&\leq 2|x_{t-1}^* - x_t^*| \\
&\leq 2|\frac{1}{t-1}\sum_{i=1}^{t-1} y_i - \frac{1}{t}\sum_{i=1}^{t} y_i| \\
&= 2|\frac{1}{t(t-1)}\sum_{i=1}^{t-1} y_i - \frac{y_t}{t}| \\
&\leq 2(\frac{1}{t(t-1)}\sum_{i=1}^{t-1} |y_i| + \frac{|y_t|}{t})
\end{aligned}$$

$$\leq \frac{2}{t} + \frac{2}{t} = \frac{4}{t}$$

Now by summing these inequalities for every $t \leq T$ we obtain:

$R_T = \sum_{i=1}^{T} \ell_i(x_i) - \sum_{i=1}^{T} \ell_i(x_T^*) \leq \sum_{i=1}^{T} \ell_i(x_i) - \sum_{i=1}^{T} \ell_i(x_i^*) \leq \sum_{i=1}^{T} \frac{4}{i} \leq 4ln(T) + 4$ which clearly means that $R_T = O(ln(T))$. Hence the sublinearity is proven and we are done.

Now why do we care about the sub-linearity of Regret? because if R_T is $o(T)$ then the average Regret factor that is $\frac{1}{T}R_T$ will go to 0 as T tends to ∞.

Chapter 2

Binary Prediction

A game of binary prediction consists of a player and an adversary. In each round the adversary chooses a bit and we have to predict it. The loss function is defined by:

$$\ell_t(x_t) = \begin{cases} 0 & x_t = y_t \\ 1 & \text{o.w} \end{cases}$$

and our objective is to minimize the cumulative loss function (which we defined in last chapter). In other words, we want to maximize our "correct guess"'s in total. But like the earlier chapter, we will need to define Regretin order to test the efficiency of our algorithm.

What is going to happen if we follow some deterministic algorithms? For instance consider the deterministic version of FTL(Follow The Leader) here. Here we will choose 0 if more than half of the previous y_i's are 0's and we will choose 1 otherwise.(Here we made the assumption that if the number of 0's and 1's are equal we will choose 1) Obviously the adversary can make our regret to be $O(T)$ where T is the number of rounds. Consider the case in which the adversary chooses the sequence $0, 1, 0, 1, \ldots$ for y_i's, the algorithm will provide the prediction sequence $1, 0, 1, 0, \ldots$ and the regret will be as follows:

$$\begin{aligned} Regret_T &= \sum_{t=1}^{T} l_t(x_t) - \min_{x\in\{0,1\}} \sum_{t=1}^{T} l_t(x) \\ &= T - [\frac{T}{2}] \\ &= O(T) \end{aligned}$$

This shows that the given FTL algorithm does not provide a sub-linear regret. This leads us to think that our ordinary deterministic algorithms do not have sub-linear regret. We need to make

sure that the adversary can not(in any way) make our Regret to be linear in terms of number of rounds(i.e. T). But could we achieve this by a deterministic algorithm?

Claim: For any deterministic algorithm and any T(number of rounds), there exist a sequence $S \in \{0,1\}^T$ for which the algorithm will have exactly T Loss.

Proof: We will build such sequence by induction on T. The base case is obvious because the player will choose either 0 or 1 and he can not choose both(!) so it is sufficient to choose the bit that he does not choose. Now imagine that we know that the statement is true for $T = i$, we will prove it's true for $T = i + 1$. The induction hypothesis tells us that there is a sequence $(y_1, ..., y_i)$, in which our algorithm will guess every y_i incorrectly. Since our algorithm is deterministic, it will choose x_{i+1} based on $(y_1, ..., y_i)$ and it will not be dependent on the value of y_{i+1}. Now, no matter what the chosen value of x_{i+1} is, we can put y_{i+1} to be $1 - x_{i+1}$. (which is the binary complement of x_{i+1}) and therefore we have built a sequence of length $i + 1$ which meets the required conditions and our induction is complete.

We want to generalize sol for instead of choosing our number deterministic-ally we choose it probability. consider $x_t \in [0, 1]$, we choose zero with probability x_t and one with probability $1 - x_t$.

Now consider a following randomized algorithm: Imagine we are in the t th round and in previous rounds, m zeros have occurred and n ones. we choose 0 with probability $\frac{m}{m+n}$ and 1 with probability $\frac{n}{m+n}$.

Even though this (FTL) algorithm seems promising to help us achieve a sub-linear regret, it does not. By providing a strategy for the adversary, we present an environment in which the regret of our algorithm would be $O(T)$.

Adversary selects zero with probability $P \in (\frac{1}{2}, 1)$ and one with probability $1 - P$. In long term $P \approx \frac{m}{m+n}$,so we choose zero with probability P. Now Lets figure out the expected value of regret.

$$\mathbb{E}[R_T] \approx (P(1-P) + (1-P)P)T - (1-P)T = (2P-1)(1-P)T$$

So the process yields the linear result.

Chapter 3

Prediction with Expert Advice

Definition: Assume we have a T-round game and N experts. Each expert has advice for our next move in each round. We should choose an expert in each round and do as he suggests. We denote $\ell_t(i)$ by the loss function if we listen to the i th expert's suggestion. So we have to choose experts $x_1, x_2, x_3, ..., x_T$ with a total loss of $\mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t)]$.

Our goal is to minimize the regret of our algorithm. We can define:

$$\text{Regret} = \mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t)] - \min_{i=1,2,..,n} \sum_{t=1}^{T} \ell_t(i) = S_T - \min_{i=1,2,..,n} S_{T,i}$$

and

$$S_{t,i} = \sum_{j=1}^{t} \ell_j(i)$$

$$S_t = \mathbb{E}[\sum_{j=1}^{t} \ell_j(x_j)]$$

Regret is the difference between the expected cost of our choices and the minimum cost of choosing one expert and acting as his advice in all T rounds. We claim we can achieve a sub-linear regret using the **Exponential Weights** algorithm:

Suppose $W_t = \sum_{i=1}^{N} e^{-\eta S_{t,i}}$. We choose expert i in round $t+1$ with probability $P_{t,i}$ which:

$$P_{t,i} = \frac{e^{-\eta S_{t,i}}}{W_t}$$

Proof: First we have that $\frac{W_T}{W_0} = \frac{W_T}{W_{T-1}} \frac{W_{T-1}}{W_{t-2}} ... \frac{W_1}{W_0}$ so:

$$\ln \frac{W_T}{W_0} = \sum_{t=1}^{T} \ln \frac{W_t}{W_{t-1}}$$

and we use this equation to prove our claim.

$$\begin{aligned}\forall_{t=1,2,\ldots,T} \ln \frac{W_t}{W_{t-1}} &= \ln \frac{\sum_{i=1}^{N} e^{-\eta S_{t,i}}}{W_{t-1}} \\ &= \ln \frac{\sum_{i=1}^{N} e^{-\eta S_{t-1,i}} e^{-\eta \ell_t(i)}}{W_{t-1}} \\ &= \ln \sum_{i=1}^{N} P_{t-1,i} e^{-\eta \ell_t(i)} \\ &\leq -\eta \mathbb{E}[\ell_t(x_t)] + \frac{\eta^2}{8} \qquad (3.1)\end{aligned}$$

and the inequality 3.1 holds because of Hoeffding's Lemma for random variable X which $a \leq X \leq b$:

$$\ln \mathbb{E}[e^{sX}] \leq s\mathbb{E}[X] + \frac{s^2(a-b)^2}{8}$$

So:

$$\begin{aligned}\ln \frac{W_T}{W_0} &\leq \sum_{t=1}^{T} (-\eta \mathbb{E}[\ell_t(x_t)] + \frac{\eta^2}{8}) \\ &= -\eta \sum_{t=1}^{T} (\mathbb{E}[\ell_t(x_t)]) + \frac{\eta^2 T}{8} \\ &= -\eta S_T + \frac{\eta^2 T}{8}\end{aligned}$$

We found an upper bound for $\ln \frac{W_T}{W_0}$. We need a lower bound to prove our claim:

$$\begin{aligned}\ln \frac{W_T}{W_0} &= \ln \sum_{i=1}^{N} e^{-\eta S_{T,i}} - \ln N \\ &\geq \ln \max_{i \in \{1,2,\ldots,N\}} e^{-\eta S_{T,i}} - \ln N \\ &= -\eta \min_{i=1,2,..,n} S_{T,i} - \ln N\end{aligned}$$

Now we have:

$$-\eta \min_{i=1,2,..,n} S_{T,i} - \ln N \leq \ln \frac{W_T}{W_0} \leq -\eta S_T + \frac{\eta^2 T}{8}$$

and:

$$\text{Regret} = S_T - \min_{i=1,2,..,n} S_{T,i} \leq \frac{\ln N}{\eta} + \frac{\eta T}{8}$$

By putting $\eta = \sqrt{\frac{8 \ln N}{T}}$ we have:

$$\text{Regret} \leq \sqrt{\frac{T \ln N}{2}}$$

and our claim has been proven.

But how exactly can this be helpful to us? Assume we have N algorithms for a certain problem and want to see which one performs better. We can see these algorithms as experts in "Prediction with Expert Advice". We now have a randomized algorithm which is expected to get the best of every algorithm we have! And how do we know that this works? because of the sub-linearity of the regret factor. In regret we test the efficiency of the algorithm by comparing it to constant algorithms, this means that the randomized algorithm has a good performance against every constant algorithm that is used in it. (the experts) Also, the way we choose the probabilities means that if a constant algorithm has better efficiency and works better, it will get chosen more because it will have less regret and a bigger probability in the long run.

We want to prove that we can not get any better order for Regret by proving that we can not get any better Regret than $O(\sqrt{T})$ in the Binary Prediction problem. Suppose we have a completely random Binary Prediction game in T rounds.

Obviously Regret $= \frac{T}{2} - \mathbb{E}[\min_{x\in\{0,1\}} \sum_{t=1}^{T} \ell_t(x)] = \frac{T}{2} - \mathbb{E}[min(X, T-X)]$ which $X \sim Binomial(T, \frac{1}{2})$. We can write $X = \sum_{i=1}^{T} X_i$ which $X_i \sim Bernoulli(\frac{1}{2})$. Assume that $Y_i = X_i - \frac{1}{2}$ and $Y = \sum_{i=1}^{T} Y_i$:

$$\begin{aligned} \text{Regret} &= \frac{T}{2} - \mathbb{E}[min(\frac{T}{2} + Y, \frac{T}{2} - Y)] \\ &= \mathbb{E}[min(Y, -Y)] \\ &= \mathbb{E}[|Y|] \end{aligned}$$

Due to CLT, we can say $Y \sim \mathcal{N}(0, c^2T)$ for some positive constant c:

$$\begin{aligned} Y \sim \mathcal{N}(0, c^2T) &\implies Y \sim c\sqrt{T}\mathcal{N}(0,1) \\ &\implies |Y| \sim c\sqrt{T}|\mathcal{N}(0,1)| \\ &\implies \mathbb{E}[|Y|] = \mathbb{E}[c\sqrt{T}|\mathcal{N}(0,1)|] = \alpha\sqrt{T} \end{aligned}$$

for positive constant $\alpha = \mathbb{E}[c|\mathcal{N}(0,1)|]$. So there is no way we can get any Regret better than $O(\sqrt{T})$.

Chapter 4

Gradient Descent

In gradient descent algorithm, we encounter several challenges. Two major problems that often arise are divergence and getting stuck in local minima. Given that the second problem of getting stuck in local minima is challenging to solve and lacks a satisfactory solution at the moment, it would be prudent to focus on the first problem. One approach to address the first problem is to introduce an additional assumption, such as considering a convex function. By assuming convexity, we can avoid the issue of getting stuck in local minima and improve the optimization process in online gradient descent. This assumption provides a more favorable landscape for finding the global minimum and can lead to better results.

Definition 4.0.1 (label=). *A In mathematics, a set $V \subset \mathbb{R}^d$ is said to be convex if for any two points x and y in V, and for any value of $\lambda \in (0,1)$, $\lambda x + (1-\lambda)y \in V$. This means that if you take any two points in a convex set, the line segment connecting them will also lie entirely within the set.*

Definition 4.0.2 (label=). *B let $f : \mathbb{R}^d \to (-\infty, +\infty)$. f is convex if epigraph of the function is convex.*

Theorem 4.0.3 (label=). *C Suppose $f : \mathbb{R}^d \to (-\infty, +\infty)$ a convex function and let $x \in dom\, f$.If f is differentiable at x then:*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \forall y \in \mathbb{R}^d$$

$$\begin{aligned} \text{Regret} &= \mathbb{E}[\sum_{t=1}^{T} \ell_t(i_t)] - \min_{1 \leq i \leq n} \sum_{t=1}^{T} \ell_t(i) \\ &= \sum_{t=1}^{T} \ell_t(x_t) - \min_{1 \leq i \leq n} \sum_{t=1}^{T} \ell_t(i) \end{aligned}$$

$$= \sum_{t=1}^{T} (\ell_t(x_t) - \ell_t(\mathbf{x}^\star))$$
$$\leq \sum_{t=1}^{T} \langle x_t - \mathbf{x}^\star, \nabla_t \rangle$$

$$\|x_{t+1} - \mathbf{x}^\star\|^2 \leq \|y_{t+1} - \mathbf{x}^\star\|^2 = \|x_t - \eta_t \nabla_t - \mathbf{x}^\star\|^2$$
$$= \|x_t - \mathbf{x}^\star\|^2 + \|\eta_t \nabla_t\|^2 - 2\eta_t \langle \nabla_t, x_t - \mathbf{x}^\star \rangle$$
$$= \langle \nabla_t, x_t - \mathbf{x}^\star \rangle \leq \frac{\|x_t - \mathbf{x}^\star\|^2 + \|\eta_t \nabla_t\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta_t}$$
$$= \frac{\|x_t - \mathbf{x}^\star\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta_t} + \frac{\eta_t \|\nabla_t\|^2}{2}$$
$$= R_t = \sum_{t=1}^{T} \left(\frac{\|x_t - \mathbf{x}^\star\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta_t}\right) + \sum_{t=1}^{T-1} \left(\frac{\eta_t \|\nabla_t\|^2}{2}\right)$$
$$\sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \|x_t - \mathbf{x}^\star\|^2 + \frac{\|x_1 - \mathbf{x}^\star\|^2}{2\eta_1} + \sum_{t=1}^{T-1} \frac{\eta_t \|\nabla_t\|^2}{2}$$

$$R_T = \sum_{t=2}^{T-1} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) D^2 + \frac{1}{2\eta_1} D^2 + \sum_{t=1}^{T-1} \frac{\eta_t G^2}{2}$$

we can assume that the functions are differentiable, and any two points within the range of the function have a bounded distance between them. Additionally, the gradient of each function is limited. These assumptions allow us to make certain conclusions and apply specific mathematical techniques when analyzing these functions.

So for summarize we have these assumption:

- functions are differentiable
- Any two points within the range of the function have a bounded distance between them
- The gradient of each function is limited

Chapter 5

Bandit

Previously we assumed that after making our prediction. we would have access to the loss function at all points, but there are many problems in which this is not true. In other words, we only have access to parts of the value of the function. In previous problems, we used our full knowledge of the loss function for minimizing Regret, for example in Experts Problem $S_{t,i}$ were updating using the loss function in all points and we used Gradient of the loss function in GD technique.

Bandit Problems: in Bandit Problems, we don't necessarily have access to full data about the loss function. Our previous algorithms won't achieve the same sub-linear Regret on these types of problems. So we have to change them using estimators of loss function to achieve our goals.

Gradient Decent in Bandit Problems:

Multi-Armed Bandit model: Assume we have a T-round game and N Bandits and we have to choose a bandit in each round t and after our choice, we see the loss $\ell_t(i)$ for our move and choosing bandit i and our goal is to achieve a sub-linear Regret.

For step $1 \leq t \leq T$, we have

$$Y_t = X_{t-1} - \eta \nabla \widehat{\ell_t}$$

$$X_t = Proj_{\omega_\epsilon}(Y_t)$$

where $\nabla \widehat{\ell_t} = (\widehat{\ell_t(1)}, \widehat{\ell_t(2)}, ..., \widehat{\ell_t(N)})$ and:

$$\widehat{\ell_{t+1}(i)} = \begin{cases} 0, & \text{if we did not choose i in round t} \\ \frac{\ell_t(i)}{X_{t,i}}, & \text{if we chose i in round t} \end{cases}$$

Also

$$\omega_\epsilon = \{X \in \mathbb{R}^N | x_i \geq \epsilon, \sum_{i=1}^{N} x_i = 1\}$$

We claim we can achieve sub-linear Regret using the GD technique with this estimation of Gradient.

Proof: Suppose $\nabla_t = \nabla \ell_t$ and $\widehat{\nabla}_t = \nabla \widehat{\ell}_t$:

$$\begin{aligned}
\text{Regret} &= \mathbb{E}[\sum_{t=1}^{T} \ell_t(i)] - \min_{1 \le i \le n} \sum_{t=1}^{T} \ell_t(i) \\
&\le \mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t) - \min_{x \in \omega_0} \sum_{t=1}^{T} \ell_t(x)] \\
&\le \mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t) - \min_{x \in \omega_\epsilon} \sum_{t=1}^{T} \ell_t(x)] + \sum_{t=1}^{T} (\ell_t(x_\epsilon^\star) - \ell_t(x_0^\star)) \\
&\le \mathbb{E}[\sum_{t=1}^{T} (\ell_t(x_t) - \ell_t(\mathbf{x}^\star))] + TN\epsilon \\
&= \text{Regret}_\epsilon + TN\epsilon \\
&\le \mathbb{E}[\sum_{t=1}^{T} \langle x_t - \mathbf{x}^\star, \nabla_t \rangle] + TN\epsilon
\end{aligned}$$

Also, we have:

$$\begin{aligned}
\|x_{t+1} - \mathbf{x}^\star\|^2 &\le \|y_{t+1} - \mathbf{x}^\star\|^2 \\
&= \|x_t - \eta \widehat{\nabla}_t - \mathbf{x}^\star\|^2 \\
&= \|x_t - \mathbf{x}^\star\|^2 + \eta^2 \|\widehat{\nabla}_t\|^2 - 2\eta \langle x_t - \mathbf{x}^\star, \widehat{\nabla}_t \rangle
\end{aligned}$$

So:

$$\begin{aligned}
\langle \widehat{\nabla}_t, x_t - \mathbf{x}^\star \rangle &\le \frac{\|x_t - \mathbf{x}^\star\|^2 + \eta^2 \|\widehat{\nabla}_t\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta} \\
&= \frac{\|x_t - \mathbf{x}^\star\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta} + \frac{\eta \|\widehat{\nabla}_t\|^2}{2}
\end{aligned}$$

And:

$$\begin{aligned}
\text{Regret}_\epsilon &\le \mathbb{E}[\sum_{t=1}^{T} \langle x_t - \mathbf{x}^\star, \nabla_t \rangle] \\
&= \mathbb{E}[\sum_{t=1}^{T} \langle x_t - \mathbf{x}^\star, \widehat{\nabla}_t \rangle] \\
&= \mathbb{E}[\sum_{t=1}^{T} (\frac{\|x_t - \mathbf{x}^\star\|^2 - \|x_{t+1} - \mathbf{x}^\star\|^2}{2\eta}) + \sum_{t=1}^{T} (\frac{\eta \|\widehat{\nabla}_t\|^2}{2})] \\
&\le \frac{\|x_1 - \mathbf{x}^\star\|^2}{2\eta} + \frac{\eta}{2} \mathbb{E}[\sum_{t=1}^{T} \|\widehat{\nabla}_t\|^2] \\
&\le \frac{1}{2\eta} + \frac{\eta TN}{2\epsilon} \\
&= \sqrt{\frac{NT}{\epsilon}}
\end{aligned}$$

So we have that Regret $\leq \sqrt{\frac{NT}{\epsilon}} + NT\epsilon$ and by putting $\epsilon = (\frac{1}{NT})^{1/3}$ we have:

$$\text{Regret} \leq (NT)^{\frac{2}{3}}$$

Prediction with Expert Advice in Bandit Problems:

Assume we have a T-round game and N choices in each round which we can see as experts, but we only see the cost of our choice after each round, not the whole loss function. Our goal is to minimize the Regret:

$$\text{Regret} = \mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t)] - \min_{i=1,2,..,n} \sum_{t=1}^{T} \ell_t(i) = S_T - \min_{i=1,2,..,n} S_{T,i}$$

where $S_{t,i} = \sum_{j=1}^{t} \ell_j(i)$ and $S_t = \mathbb{E}[\sum_{j=1}^{t} \ell_j(x_j)]$.

In this case, we use an estimator of $\ell_t(i)$ by using the Exponential-weight algorithm for Exploration and Exploitation.

EXP3 Algorithm: Suppose $\widehat{S_{t,i}} = \sum_{j=1}^{t} \widehat{\ell_j(i)}$. In each step $t = 0, 1, ..., T-1$, denote $W_t = \sum_{i=1}^{N} e^{-\eta \widehat{S_{0,i}}}$ and for $i = 1, 2, ..., N$ we have

$$P_{t,i} = \frac{e^{-\eta \widehat{S_{t,i}}}}{W_t}$$

In round $t+1$ we choose expert i with probability of $P_{t,i}$ and we put

$$\widehat{\ell_{t+1}(i)} = \begin{cases} 0, & \text{if we did not choose i in round t} \\ \frac{\ell_t(i)}{P_{t,i}}, & \text{if we chose i in round t} \end{cases}$$

By using this estimator we have that :

$$\mathbb{E}[\widehat{\ell_t(i)}] = \ell_t(i)$$

Proof: First we have that $\frac{W_T}{W_0} = \frac{W_T}{W_{T-1}} \frac{W_{T-1}}{W_{t-2}} ... \frac{W_1}{W_0}$ so $\ln \frac{W_T}{W_0} = \sum_{t=1}^{T} \ln \frac{W_t}{W_{t-1}}$. So:

$$\begin{aligned} \forall_{t=1,2,...,T} \ln \frac{W_t}{W_{t-1}} &= \ln \frac{\sum_{i=1}^{N} e^{-\eta \widehat{S_{t,i}}}}{W_{t-1}} \\ &= \ln \frac{\sum_{i=1}^{N} e^{-\eta S_{t-1,i}} e^{-\eta \widehat{\ell_t(i)}}}{W_{t-1}} \\ &= \ln \sum_{i=1}^{N} P_{t-1,i} e^{-\eta \widehat{\ell_t(i)}} \\ &\leq \ln \sum_{i=1}^{N} P_{t-1,i} (1 - \eta \widehat{\ell_t(i)} + (\eta \widehat{\ell_t(i)})^2/2) \end{aligned} \tag{5.1}$$

$$
\begin{aligned}
&= \ln\left(1 - \eta \sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2\right) \\
&\leq \ln e^{-\eta\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2} \qquad (5.2) \\
&= -\eta \sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2
\end{aligned}
$$

and the inequality 5.1 holds because of $e^{-x} \leq 1 - x + \frac{x^2}{2}$ and inequality 5.2 holds because of $1 - x \leq e^{-x}$. So:

$$
\begin{aligned}
\ln\frac{W_T}{W_0} &\leq \sum_{t=1}^{T}\left(-\eta\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2\right) \\
&= -\eta\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2
\end{aligned}
$$

We found an upper bound for $\ln\frac{W_T}{W_0}$.

We need a lower bound for $\ln\frac{W_T}{W_0}$ in order to prove our claim:

$$
\begin{aligned}
\ln\frac{W_T}{W_0} &= \ln\sum_{i=1}^{N} e^{-\eta\widehat{S_{T,i}}} - \ln N \\
&\geq \ln\max_{i\in\{1,2,\ldots,N\}} e^{-\eta\widehat{S_{T,i}}} - \ln N \\
&= -\eta\min_{i=1,2,..,n}\widehat{S_{T,i}} - \ln N
\end{aligned}
$$

Now we have:

$$
-\eta\min_{i=1,2,..,n}\widehat{S_{T,i}} - \ln N \leq \ln\frac{W_T}{W_0} \leq -\eta\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta^2}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2
$$

$$
\begin{aligned}
&\implies -\min_{i=1,2,..,n}\widehat{S_{T,i}} - \frac{\ln N}{\eta} \leq -\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2 \\
&\implies \sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} - \min_{i=1,2,..,n}\widehat{S_{T,i}} \leq \frac{\ln N}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2 \\
&\implies \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)} - \min_{i=1,2,..,n}\widehat{S_{T,i}}\right] \leq \mathbb{E}\left[\frac{\ln N}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2\right] \\
&\implies \text{Regret} = S_T - \min_{i=1,2,..,n} S_{T,i} \leq \frac{\ln N}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2\right]
\end{aligned}
$$

We have that $\mathbb{E}[P_{t-1,i}\widehat{\ell_t(i)}^2] = P_{t-1,i}{}^2(\frac{\ell_t(i)}{P_{t-1,i}})^2 = \ell_t(i)^2 \leq 1$, so:

$$\begin{aligned}
\text{Regret} &\leq \frac{\ln N}{\eta} + \frac{\eta}{2}\mathbb{E}[\sum_{t=1}^{T}\sum_{i=1}^{N} P_{t-1,i}\widehat{\ell_t(i)}^2] \\
&\leq \frac{\ln N}{\eta} + \frac{\eta}{2}\mathbb{E}[\sum_{t=1}^{T}\sum_{i=1}^{N} 1] \\
&= \frac{\ln N}{\eta} + \frac{NT\eta}{2}
\end{aligned}$$

By putting $\eta = \sqrt{\frac{2\ln N}{NT}}$ we have:

$$\text{Regret} \leq \sqrt{\frac{TN\ln N}{2}}$$

and we achieved a sub-linear Regret.

Chapter 6

Contextual Bandit

6.1 Re-state EXP3 Algorithm based on Reward

So far, we've looked at adversarial bandit. The adversarial bandit problem:

- For rounds $t = 1, 2, ..., n$:
  - Learner selects distribution $p_t \in P_k$ and samples A_t from p_t.
  - Learner observes reward $X_t = x_{tA_t}$.

Here we define the problem with reward instead of using loss. So each action in every round has a certain reward and we only see the reward of the action we choose.(limited feedback) Here regret is defined as below: $R_n(\pi, x) = \max_{i\in[k]} \sum_{t=0}^{n} x_{ti} - \mathbb{E}[\sum_{t=1}^{n} x_{tA_t}]$ where A_t is the random variable that is chosen by our policy.(i.e. π) Here we try to control the maximum regret factor that is defined by $R_n^*(\pi) = \sup_{x\in[0,1]^{n\times k}} R_n(\pi, x)$. The name adversarial comes from this definition because somehow we are considering the worst case scenario for our policy and try to find algorithms that have small regrets in worst case scenarios. Now that we have changed our vision towards the problem by replacing loss with reward, we want to take another look to the changed version of the EXP-3 algorithm:

- Input: n, k, η
- $\hat{S}_{0i} = 0$
- for $t = 1, 2, ..., n$ do:
  - calculate the sampling distribution $P_t : P_{t,i} = \frac{exp(\eta \hat{S}_{t-1,i})}{\sum_j exp(\eta \hat{S}_{t-1,j})}$

  - sample $A_t \sim P_t$ and observe reward X_t
  - calculate $\hat{S}_{t,i} : \hat{S}_{t,i} = \hat{S}_{t-1,i} + (1 - \frac{I\{A_t=i\}(1-X_t)}{P_{ti}})$

- end loop

where $1 - \frac{I\{A_t=i\}(1-X_t)}{P_{ti}}$ is our estimate for the reward of each action in round t. This means that the reward will be 1 for each action that is not the chosen action and the reward of the chosen action will be equal to $1 - \frac{1-X_t}{P_{ti}}$ so that our estimate will be unbiased.

6.2 Contextual Bandit

Now we define the contextual bandit problem. Here we have some contextual background information. Here we consider the contextual data to be helpful because if we don't use the data, we can use the previous methods. So by having the data we should design algorithms with lower regrets. For instance, we want to design an algorithm which recommends movies to the user based on the background information that we have about the user. This exactly refers to an online process because the data of the user is available to us only in different periods of time(e.g. movies he/she has watched in previous month) but how can we define the problem with mathematical notions so that the background information is well-defined? Contextual bandit problem:

- Adversary secretly chooses rewards $(x_t)_{t=1}^n$ with $x_t \in [0,1]^k$
- Adversary secretly chooses contexts $(c_t)_{t=1}^n$ with $c_t \in \mathcal{C}$
- For each round $t = 1, 2, ..., n$:
  - Learner observes $c_t \in \mathcal{C}$
  - Learner selects distribution $p_t \in P_{k-1}$ and samples A_t from P_t
  - Learner observes reward $X_t = x_{tA_t}$
- end loop

We should choose c_i's carefully and assign similar choices with the same context. But how do we define the regret factor? $R_n(\pi, x, c) = \mathbb{E}[\sum_{c\in\mathcal{C}} \max_{i\in[k]} \sum_{t\in[n]:c_t=c}(x_{ti} - X_t)]$ and $R_n^*(\pi) = \sup_{x\in[0,1]^{n\times k}, c\in\mathcal{C}} R_n(\pi, x, c)$ By this definition we made the problem stronger. In other words, previously, we were only considering the case where we only have one single context. Now we have stronger opponents for our algorithm and we can examine the algorithm's effectiveness with respect

to stronger algorithms because we are considering the constant choice for every context. But how can we achieve a good regret bound on this problem? Since by our definition we are coloring each choice with some context and the regret is defined separately for each context, we can run the previous algorithms for each context separately in order to achieve a bound on regret. So we run the EXP-3 algorithm on every single context. By doing this we see: $R_{n,c} \leq 2\sqrt{k \log(k) \sum_{t=1}^{n} I\{c_t = c\}}$. In other words, since we know that the upper bound for EXP-3 algorithm is $2\sqrt{k \log(k) n}$, it's only enough to count the number of occurrences in each context. So the total regret factor has the upper bound: $R_n = \sum_{c \in \mathbb{C}} R_{nc} \leq 2\sqrt{k \log(k)} \times (\sum_{c \in \mathbb{C}} \sqrt{\sum_{t=1}^{n} I\{c_t = c\}})$. By Arithmetic Mean-Square Mean inequality we have that this expression is maximized when the number of occurrences in each context is constant. So we have $2\sqrt{k \log(k)} \times (\sum_{c \in \mathbb{C}} \sqrt{\sum_{t=1}^{n} I\{c_t = c\}}) \leq 2\sqrt{k \log(k)} \times |\mathbb{C}| \times \sqrt{\frac{n}{|\mathbb{C}|}} = 2\sqrt{k \log(k) |\mathbb{C}| n}$. If we ignore the context, we would have the same upper bound in EXP-3 which is better but as we mentioned before, here we are examining the algorithms effectiveness towards better opponents so it is logical that we obtain a regret factor worse than before. This bound works fine for large n's but it doesn't have the same effectiveness while applying it to small n's.(If $n \leq 4k \log(k) |\mathbb{C}|$, the upper bound is obvious). Here we can look at the result in another way. We can say that if we group similar rounds into the same contexts, then $\sum \max \sum x_{ti}$ will have more value and then we can say that the upper bound is a good bound. This result is good but is it satisfying enough for us? We need to present a more general framework for the problem. This framework will be defined by focusing on the set of contexts and different mapping of contexts to the actions. We define the Regret factor in another way: $R_n = \mathbb{E}[\max_{\phi} \sum_{t=1}^{n} (x_{t\phi(c_t)} - X_t)]$ where $\phi : \mathcal{C} \mapsto [k]$ is a arbitrary function. In other words, we are considering the best possible mapping which is the best opponent for our algorithm. Here we can make ϕ to be arbitrary or we can select it from a set of well-defined functions. We present some cases of these sets:

- the set of all functions
- for a specific partition of $\mathcal{C}$ consider all functions that map each subset of the partition to a constant number
- Good candidates. We can choose good candidates for the functions by other algorithms such as non-online algorithms and then do our algorithm for those candidates: $\phi_1, \phi_2, ..., \phi_M : \mathcal{C} \mapsto [k], \Phi = \{\phi_1, \phi_2, ..., \phi_M\}$

Now how can we solve the problem with this setting?(Not having every single function in Φ) We can see all of these functions as a single expert who is trying to suggest some action in every round.

But can we run the prediction with expert advice with no problem at all? Here we can not use that algorithm because the problems settings are different. Here we have limited feedback.(unlike what we based our algorithm on in prediction with expert advice) Now we try to present a solution for this setting. First, just like before, we try using random algorithms. In other words, instead of choosing a single action based on our data, we try selecting a distribution on actions. So the general setting will be changed to this: $\Phi = \{\phi_1, \phi_2, ..., \phi_M\}, \phi_i : \mathcal{C} \mapsto P_k$
$R_n = \mathbb{E}[\max_{\phi \in \Phi} \sum_t (\sum_{i=1}^k (\phi(c_t)_i \times x_{t,i} - X_t))]$
where $\phi(c_t)_i$ is the probability of choosing ith action in round t. Now we want to define the problem properly based on this setting:

- Adversary secretly chooses rewards $x \in [0, 1]^{n \times k}$
- Every expert secretly chooses predictions $E^{(1)}, E^{(2)}, ..., E^{(n)}$
- For rounds $t = 1, 2, ..., n$:
  - Learner observes predictions of all experts, $E^{(t)} \in [0, 1]^{M \times k}$, where M is the number of experts
  - Learner selects a distribution $p_t \in \mathbb{P}_{k-1}$
  - Action A_t is sampled from p_t and the reward is $X_t = x_{tA_t}$
- end loop

The new Regret will be defined as:
$R_n = \mathbb{E}[\max_{m \in [M]} \sum_{t=1}^n (E_m^{(t)} x_t - X_t)]$
where $E_m^{(t)}$ refers to the distribution that is presented by the mth expert on actions in round t. $E_m^{(t)}$ is the mth row of $E^{(t)}$. Now we want to present a solution for this setting. The basic idea is to use the idea in prediction with expert advice. We sample A_t $p_t = Q_t E^{(t)}$ where Q_t refers to the distribution on experts choices. Now we have to choose a proper distribution based on rewards in previous rounds. But because of the limited feedback property, we need to have a good estimation of other action's rewards in each round. In EXP-3, we used the estimation $\hat{X}_{ti} = 1 - \frac{I\{A_t = i\}}{P_{ti} + \epsilon}(1 - X_t)$, how can we use this to our advantage here? We use the estimation $\tilde{X}\check{X}_t = E_{(t)}\hat{X}_t$, where $\check{X}$ refers to a vector in which the ith element is the expected of reward while listening to the ith expert. Now we use the exact same trick in prediction with expert advice algorithm and listen to the mth expert with weight $\exp(\eta \tilde{X}_{t,m})$ where $\check{X}_{t,m}$ refers to the sum of mth elements of $\check{X}_1, \check{X}_2, ..., \check{X}_t$. In other words, it is the expected reward of the mth expert up to the tth round.

Bounding the regret in this algorithm:

Lemma 6.2.1 (label=)**.** *For each $m^* \in [M]$,*

$$\sum_{t=1}^{n} \check{X}_{t,m^*} - \sum_{t=1}^{n}\sum_{m=1}^{M} Q_{tm}\check{X}_{tm} \leq \frac{\log(M)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{n}\sum_{m=1}^{M} Q_{tm}(1-\check{X}_{tm})^2$$

If we prove the lemma, by getting expected from both sides we will have: $R_n \leq \frac{\log(M)}{\eta} + \frac{\eta}{2}\mathbb{E}[\sum\sum Q_{tm}(1-\check{X}_{tm})^2]$. Now if we bound the left-hand-side's expected factor, we are done.
Now we have:
$\check{X}_{tm} = \sum E^{(t)}_{mi}\hat{X}_{ti} = \sum E^{(t)}_{mi}(1 - \frac{I\{A_t=i\}(1-X_t)}{p_{ti}}) = 1 - E^{(t)}_{mA_t} \times \frac{1-X_{tA_t}}{p_{tA_t}}$ so we can write:
$\mathbb{E}_t[(1-\check{X}_{tm})^2] = \mathbb{E}_t[(\frac{E^{(t)}_{mA_t}(1-x_{tA_t})}{p_{tA_t}})^2] = \sum_{i=1}^{k}\frac{(E^{(t)}_{mi}(1-x_{ti}))^2}{p_{ti}} \leq \sum_{i=1}^{k}\frac{E^{(t)}_{mi}}{p_{ti}}$. Now that we have this, we can bound the left-hand-side's expected:
$\mathbb{E}[\sum_{m=1}^{M} Q_{tm}(1-\check{X}_{tm})^2] \leq \sum_{m=1}^{M} Q_{tm}\mathbb{E}[\sum\frac{E^{(t)}_{mi}}{p_{ti}}] = \mathbb{E}[\sum_{i=1}^{k}\frac{\sum_{m=1}^{M} Q_{tm}E^{(t)}_{mi}}{p_{ti}}] = \mathbb{E}[\sum_{i=1}^{k} 1] = k.$
So if we sum this for every round we have:
$R_n \leq \frac{\log(M)}{\eta} + \frac{\eta}{2}nk = O(\sqrt{nk\log(M)})$. So by choosing the right value for η, the algorithm will have sub-linear regret. Now we want to apply this bound to a specific Φ. If we consider Φ to be all functions from $\mathbb{C}$ to $[k]$, we have $M = k^{|\mathbb{C}|}$. So we can have $R_n \leq \sqrt{nk\log\left(k^{|\mathbb{C}|}\right)} = \sqrt{nk|\mathbb{C}|\log(k)}$ for regret. This upper bound is the exact upper bound we could achieve, had we run EXP-3, $|\mathbb{C}|$ times, once for every context! In general, if we choose any set of functions, we can have the upper bound $\sqrt{nk\log(|\Phi|)}$ for the regret factor.

Chapter 7

Linear Bandit

In this chapter, we are going to see another type of bandit problems. The Linear Bandit problem will be defined as below:

- Input:d, A
- For rounds $t = 1, ..., n$:
  - Adversary secretly chooses a vector $y_t \in \mathbb{R}^d$.
  - Learner chooses a prediction vector $A_t \in A$
  - Learner observes loss equal to $< A_t, y_t >$
- end loop

We can see the adversary bandit problem as a case of this problem. It is sufficient to put $A = \{e_1, e_2, ..., e_k\}$ and y_k can be interpreted as the vector which the ith element in it is equal to the loss while taking the ith action. Here we need to add some constraint on the loss function.(much like what we did in adversarial bandit) So we add the constraint: $y_t \in \mathrm{L} = \{x \in \mathbb{R}^d : \sup_{a \in A} |< a, x >| \leq 1\}$. So in each round, our loss is at most equal to 1. The regret factor will be defined pretty much like before, So we will not write any new equations defining regret. Now that we have defined the problem properly, we need solutions of sub-linear regret for it. Here we can use the algorithm that was presented in chapter 5 as Bandit Gradient Descent because the loss function is convex and has the necessary properties. So there is an algorithm with Regret = $O(n^{\frac{2}{3}} D^{\frac{3}{2}})$. But we will present one with even better regret bound! The basic idea is to use the previous ideas in EXP-3. So we are going to do:

- For rounds $t = 1, 2, ..., n$:

- Choose distribution p_t on A
- Sample action A_t from p_t
- Observe loss $Y_t =< A_t, y_t >$

And we want to choose exponential weights for loss functions based on our estimation of the expected loss of every constant action up to the tth round. So it is logical to put $\check{p}_t(a) \sim exp(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a))$. But this will not be sufficient. So we will change this a bit by adding an exploration factor to weights: $p_t = (1-\lambda)\check{p}_t(a) + \lambda\pi(a)$, in which π represents a distribution on A and is added for exploration. Now we have presented the way to choose the weights but we still do not have a good estimation for $Y_t(a)$. In order to achieve this, we want to use a linear estimation on Y_t which will be equal to $\hat{Y}_t = R_t A_t Y_t$. We will choose R_t based on our needs. Here we can write:
$\mathbb{E}[\hat{Y}_t] = \mathbb{E}[R_t A_t A_t^T y_t] = R_t \mathbb{E}[A_t A_t^T] y_t = R_t(\sum_{a\in A} p_t(a) a a^T) y_t$. Now if we want the estimator to be unbiased, we should have that the expected value of the estimator for each possible y_t is y_t. So we should have $R_t(\sum_{a\in A} p_t(a) a a^T) = I$. So we will put $R_t = (\sum_{a\in A} p_t(a) a a^T)^{-1}$. But one thing we should consider is the possibility of $(\sum_{a\in A} p_t(a) a a^T)$ not having an inverse. We know that aa^T is a positive semi-definite matrix And the sum of some positive semi-definite matrices will be a positive semi-definite matrix. Therefore, we have $(\sum_{a\in A} p_t(a) a a^T)$ is positive semi-definite. So if we assume that the span of aa^T's would be all vectors in $\mathbb{R}^d$, we will have that $(\sum_{a\in A} p_t(a) a a^T)$ will have an inverse.

EXP-3 Algorithm for linear bandit problem:

- For rounds $t = 1, 2, ..., n$:
  - Compute the distribution $p_t(a) = \lambda\pi(a) + (1-\lambda)\frac{\exp(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a))}{\sum_{a'\in A} \exp(-\eta \sum_{s=1}^{t-1} \hat{Y}_s(a'))}$
  - $A_t \sim p_t$
  - Observe loss $Y_t =< A_t, y_t >$ and compute loss estimate: $\hat{Y}_t = Q_t^{-1} A_t Y_t$ and $\hat{Y}_t(a) =< a, \hat{Y}_t >$
- end loop

Chapter 8

Delay and Cooperation in Non-stochastic Linear Bandits

8.1 Preliminaries

An integrable function $f : \mathbb{R}^m \to \mathbb{R}_+$ is a density function if

$$\int_{\mathbb{R}^n} f(x)\, dx = 1.$$

Every non-negative integrable function f gives rise to a probability measure on the measurable subsets of $\mathbb{R}^m$ defined by

$$\text{Prob}(S) = \frac{\int_S f(x)\, dx}{\int_{\mathbb{R}^m} f(x)\, dx.}$$

Definition 8.1.1. *We list some notions as follows:*

1. *The convex hull of a set of points $\mathcal{A}$ in a Euclidean space is denoted as $conv(\mathcal{A})$ and is defined by:*
$$conv(\mathcal{A}) = \left\{ \sum_{i=1}^{n} \lambda_i x_i \mid x_i \in X, \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

2. *$Sym(m)$ represents the set of symmetric $m \times m$ matrices.*

3. *Let $A \in Sym(m)$ be a positive semidefinite matrix. For any vector $x \in \mathbb{R}^m$), norm of x respect to A in denoted by*
$$\|x\|_A = \|A^{\frac{1}{2}} x\|_2 = \sqrt{x^\top A x}$$

4. *Let $X = (X_1, \ldots, X_m)$ is a random variable distributed according to p on $\mathbb{R}^m$, then its mean value is a vector in $\mathbb{R}^m$ and is given by:*

$$\mathbb{E}[X] = (\mathbb{E}[X_1], \mathbb{E}[X_2], \ldots, \mathbb{E}[X_m]) .$$

Note that the mean value of distribution p is denoted by $\mu(p)$ and equal to

$$\mu(p) = \mathbb{E}_{x\sim p}[x].$$

5. *The covariance matrix associated with the distribution p in $\mathbb{R}^m$ is denoted as $S(p) \in Sym(m)$. The covariance matrix $S(p)$ is given by:*

$$S(p) = \begin{bmatrix} cov(X_1, X_1) & cov(X_1, X_2) & \cdots & cov(X_1, X_m) \\ cov(X_2, X_1) & cov(X_2, X_2) & \cdots & cov(X_2, X_m) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_m, X_1) & cov(X_m, X_2) & \cdots & cov(X_m, X_m) \end{bmatrix}.$$

As a result,

$$S(p) = \mathbb{E}_{x\sim p}[xx^\top].$$

8.1.1 Mahalanobis distance

The Euclidean distance measures the straight-line distance between two points in a space. However, in multivariate datasets where variables are correlated, Euclidean distance may not adequately capture the "true" distance between points. Let p is a distribution on $\mathcal{A} \subseteq \mathbb{R}^m$. The Mahalanobis distance is a measure of the distance between a point $x \in \mathcal{A}$ and the distribution p of points on $\mathcal{A}$, considering the covariance structure of the data Mahalanobis (1936). It is defined as follows:

$$D_M(x) = \sqrt{(x - \mu(p))^\top \mathbf{S}^{-1}(p)(x - \mu(p))}.$$

where $D_M(\mathbf{x})$ is the Mahalanobis distance of the observation x. Note that

1. The Mahalanobis distance adjusts for the covariance structure, providing a more meaningful measure in multivariate datasets.

2. Mahalanobis distance normalizes the difference between the point and the mean by the covariance matrix, making the metric invariant to linear transformations.

3. A smaller Mahalanobis distance indicates that the point is closer to the center of the distribution, considering the covariance structure.

4. Mahalanobis distance is often used in outlier detection, where points with unusually large distances may be considered outliers.

A player is given with the number T of rounds and an action set $\mathcal{A} \subseteq \mathbb{R}^m$ before the game start. The action set $\mathcal{A}$ is an arbitrary compact set in $\mathbb{R}^m$ which is not contained in any proper linear subspace. Let origin coordinate of $\mathbb{R}^m$ is $\mu(p)$, then $D_M(x) = \|x\|_{S^{-1}(p)}$.

8.1.2 Distribution trancation

Definition 8.1.2. *Let p is distribution on $B = conv(\mathcal{A})$ with mean value $\mu(p) \in \mathbb{R}^m$. Define a truncated distribution p' by*

$$p'(x) = \frac{p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} \leq m\gamma^2\}}{Prob_{y\sim p}[\|y\|^2_{S^{-1}(p)} \leq m\gamma^2]}.$$

where $\gamma \geq 4\log(10mT)$ is a parameter which is called truncation level.

Note that The parameter γ is a threshold or truncation level. If the norm of a point x respect to $S^{-1}(p)$ is greater than $m\gamma^2$, the probability of x under the truncated distribution is set to zero. That means $\|x\|^2_{S(p)^{-1}} > m\gamma^2$ would be truncated to have zero probability since x located "farther" from the origin, $\mu(p)$, as measured by the norm. As a result, the choice of γ determines when a point is considered "far" from the origin based on this norm.

8.1.3 Log-Concave Distributions

Definition 8.1.3. *A function $f(x)$ is concave over an interval if, for any two points x_1 and x_2 in that interval and for any λ in the range $[0, 1]$, the following inequality holds:*

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda)f(x_2).$$

Definition 8.1.4. *Let $p : \mathcal{B} \to \mathbb{R}_{\geq 0}$ be a density function then $\log(p(x))$ is a concave function. Then its probability distribution called a* log-concave distribution.

Note that a function $f : \mathbb{R}^m \to \mathbb{R}_+$ is called *log-concave* if it satisfies

$$f(\alpha x + (1-\alpha)y) \geq f(x)^\alpha f(y)^{1-\alpha}$$

for every $x, y \in \mathbb{R}^m$, and $0 \leq \alpha \leq 1$.

Lemma 8.1.5. *Lovász and Vempala (2007) Let X be a random point drawn from a log-concave distribution on $\mathbb{R}$. Assume that $\mathbb{E}[X^2] \leq 1$. Then for every $t > 1$,*

$$Prob(|X| > t) < e^{-t+1}.$$

Theorem 8.1.6. *Lovász and Vempala (2007) All marginals as well as the distribution function of a log-concave function are log-concave. The convolution of two log-concave functions is log-concave.*

Lemma 8.1.7. *If x follows a log-concave distribution p over $\mathbb{R}^m$ satisfying $S(p) \preceq I$, we have*

$$Prob\left[\|x\|^2_{S(p)^{-1}} \geq m\alpha^2\right] \leq m\, exp(1-\alpha) \tag{8.1}$$

for an arbitrary $\alpha > 0$.

Proof. Since p is a log-concave distribution, by Theorem8.1.6, all marginals are log-concave. We have $E[x_i^2] \leq 1$ becasue $S(p) \preceq I$. Hence, by union bound and Lemma 8.1.5, we have

$$\text{Prob}\left[\|x\|_2^2 \geq m\alpha^2\right] \leq \text{Prob}\left[\exists i \in [m] : x_i^2 \geq \alpha^2\right] \leq \sum_{i=1}^{m} \text{Prob}\left[|x_i| \geq \alpha\right] \leq m\exp(1-\alpha).$$

□

Lemma 8.1.8. *Suppose that p is a log-concave distribution over $\mathcal{B}$. For any function $f : \mathcal{B} \to [-1, 1]$ and $\gamma \geq 4\log(10mT)$, we have*

I. $\left|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim p'}[f(x)]\right| \leq \frac{1}{T}$.

II. $\frac{T}{T+1} \cdot S(p) \preceq S(p') \preceq \frac{T+1}{T} \cdot S(p)$.

Proof. Part I By definition of p', we have

$$\begin{aligned}\mathbb{E}_{x\sim p'}[f(x)] &= \int_{x\in\mathcal{B}} f(x)p'(x)\,dx \\ &= \frac{\int_{x\in\mathcal{B}} f(x)\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} \leq m\gamma^2\}\right)\,dx}{\text{Prob}_{y\sim p}\left[\|y\|^2_{S^{-1}(p)} \leq m\gamma^2\right]}.\end{aligned}$$

Let $\delta = \text{Prob}_{y\sim p}\left[\|y\|^2_{S^{-1}(p)} > m\gamma^2\right]$, then

$$\begin{aligned}\mathbb{E}_{x\sim p'}[f(x)] &= \frac{1}{1-\delta}\int_{x\in\mathcal{B}} f(x)\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} \leq m\gamma^2\}\right)\,dx \\ &= \frac{1}{1-\delta}\left(\mathbb{E}_{x\sim p}[f(x)] - \int_{x\in\mathcal{B}} f(x)\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right)\,dx\,dx\right), \quad (8.2)\end{aligned}$$

Thus,

$$\begin{aligned}
&\left|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim p'}[f(x)]\right| \\
&= \left|\mathbb{E}_{x\sim p}[f(x)] - \frac{1}{1-\delta}\left(\mathbb{E}_{x\sim p}[f(x)] - \int_{x\in\mathcal{B}} f(x)\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right) dx\, dx\right)\right| \\
&= \frac{1}{1-\delta}\left|-\delta\mathbb{E}_{x\sim p}[f(x)] + \int_{x\in\mathcal{B}} f(x)\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right) dx\right|
\end{aligned}$$

Since $f : \mathcal{B} \to [-1, 1]$, we have

$$\begin{aligned}
&\left|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim p'}[f(x)]\right| \\
&\le \frac{1}{1-\delta}\left(\delta\mathbb{E}_{x\sim p}[1] + \int_{x\in\mathcal{B}} \mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}p(x)\, dx\right) \\
&= \frac{2\delta}{1-\delta},
\end{aligned} \tag{8.3}$$

Now, we try to estimate $\delta = \mathrm{Prob}_{x\sim p}\left[\|x\|^2_{S^{-1}(p)} > m\gamma^2\right]$. Since $\|x\|^2_{S(p)^{-1}} = \|S(p)^{-\frac{1}{2}}x\|_2^2$, we try to verify the assumptions of Lemma 8.1.7 for $S(p)^{-\frac{1}{2}}x$.

- The random variable $S(p)^{-1/2}x$ follows a log-concave function p.
- The covariance of $S(p)^{-1/2}x$ is equal to

$$\mathbb{E}\left[S(p)^{-1/2}xx^TS(p)^{-1/2}\right] = S(p)^{-1/2}\mathbb{E}\left[xx^T\right]S(p)^{-1/2} = S(p)^{-1/2}S(p)S(p)^{-1/2} = S(p)$$

by Lemma 8.1.7 and $\gamma \ge 4\log(10mT)$, we obtain

$$\delta \le m\exp(1-\gamma) \le 3m\exp(-\gamma) \le \frac{3m}{(10mT)^4} \le \frac{1}{6T}. \tag{8.4}$$

By inequations 8.3 and 8.4, we have

$$\left|\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim p'}[f(x)]\right| \le \frac{2}{6T-1} \le \frac{1}{3T-1/2} \le \frac{1}{T}.$$

Part II. To compare $S(p')$ and $S(p)$, we use the quadratic form $y^\top S(p')y$ and then by 8.2, we have

$$\begin{aligned}
y^TS(p')y &= y^\top\mathbb{E}_{x\sim p'}\left[xx^\top\right]y \\
&= \mathbb{E}_{x\sim p'}\left[y^\top xx^\top y\right] \\
&= \frac{1}{1-\delta}\left(\mathbb{E}_{x\sim p}[y^\top xx^\top y] - \int_{x\in\mathcal{B}} y^\top xx^\top y\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right) dx\right), \\
&= \frac{1}{1-\delta}\left(y^TS(p)y - \int_{x\in\mathcal{B}} y^\top xx^\top y\ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right) dx\right) \\
&\le \frac{1}{1-\delta}y^TS(p)y
\end{aligned} \tag{8.5}$$

Since $\frac{1}{1-\delta} \leq \frac{T+1}{T}$, we have

$$y^\top S(p')y \leq \frac{T+1}{T} y^\top S(p)y. \tag{8.6}$$

Since Inequality 8.6 holds for all $y \in \mathbb{R}^m$, we obtain

$$S(p') \leq \frac{T+1}{T} S(p).$$

Furthermore, we have

$$\begin{aligned}
y^\top S(p)y - y^\top S(p')y = y^\top S(p)y - & \\
& \frac{1}{1-\delta}\left(y^T S(p)y - \int_{x\in\mathcal{B}} y^\top xx^\top y \ \left(p(x)\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right) dx\right) \\
&= -\frac{\delta}{1-\delta} y^\top S(p)y - \frac{1}{1-\delta}\mathbb{E}_{x\sim p}\left[(y^\top xx^\top y\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right] \\
&= -\frac{\delta}{1-\delta}\mathbb{E}_{x\sim p}\left[(y^\top xx^\top y\right] + \frac{1}{1-\delta}\mathbb{E}_{x\sim p}\left[(y^\top xx^\top y\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right] \\
&= \mathbb{E}_{x\sim p}\left[(y^\top xx^\top y\right] - \frac{1}{1-\delta}\mathbb{E}_{x\sim p}\left[y^\top xx^\top y\mathbb{I}\{\|x\|^2_{S^{-1}(p)} \leq m\gamma^2\}\right] \\
&\leq \mathbb{E}_{x\sim p}\left[y^\top xx^\top y\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right] \\
&\leq y^T S(p)y\mathbb{E}_{x\sim p}\left[\|x\|^2_{S^{-1}(p)}\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right],
\end{aligned}$$

Since for every two random variable X and Y, we have

$$\mathbb{E}[XY] \leq \mathbb{E}[X]\mathbb{E}[Y].$$

Thus

$$E_{x\sim p}\left[y^\top xx^\top y\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right] \leq \mathbb{E}_{x\sim p}\left[y^\top xx^\top y\right] E_{x\sim p}\left[\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right]$$

Thus

$$y^\top S(p)y - y^\top S(p')y \leq y^\top S(p)y E_{x\sim p}\left[\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right].$$

By Lemma 8.1.7, we have

$$\begin{aligned}
\mathbb{E}_{x\sim p}\left[\|x\|^2_{S^{-1}(p)}\mathbb{I}\{\|x\|^2_{S^{-1}(p)} > m\gamma^2\}\right] &= \text{Prob}\left(\|x\|^2_{S(p)-1} > m\gamma^2\right) = \text{Prob}\left(\|S(p)-1x\|^2_2 > m\gamma^2\right) \\
&\leq \frac{3m}{(10mT)^4} \\
&\leq \frac{1}{2T},
\end{aligned} \tag{8.7}$$

Thus, we have

$$y^\top S(p)y - y^\top S(p')y \leq \frac{1}{2T} y^T S(p)y.$$

So

$$\frac{T}{T+1}y^\top S(p)y \le \frac{2T-1}{2T}y^\top S(p)y \le y^\top S(p')y.$$

We have

$$\frac{T}{T+1}S(p) \le S(p').$$

□

Lemma 8.1.9. *If y follows a one-dimensional log-concave distribution such that $\mathbb{E}[y^2] \le s^2 \le 1/100$, we have*

$$\mathbb{E}[g(y)] \le s^2 + 30exp\left(-\frac{1}{s}\right)\sqrt{2s^2}.$$

Proof. Since y^2 is log-concave distribution and

$$\mathbb{E}[\frac{y^2}{s^2}] \le 1.$$

Thus by Lemma 8.1.7, for each $n \in \mathbb{N}$, we have

$$\text{Prob}\left[\frac{y}{-}\right]$$

□

8.2 Problem Settings

1. A player is given the number T of rounds.
2. An compact action set $\mathcal{A} \subseteq \mathbb{R}^m$.
3. In each round $t \in [T]$,
   - the player choose action a_t.
   - if $t > d$, the enviroment reveals the loss $\ell^\top a_{t-d} \in \mathbb{R}$.
   - Without loss of generality, we assume $d \le T-1$.
   - It is assumed that $|\ell_t^\top a| \le 1$ for all $a \in \mathcal{A}$.
   - it is assumed that the sequence $\{\ell_t\}_{t=1}^T$ is an arbitrary non-adaptive sequence. That means each ℓ_t is not to depend on the output of the algorithm.
   - $R_T = \sum_{t=1}^T \ell_t^\top a_t - \min_{a \in \mathcal{A}}^* \sum_{t=1}^T \ell^\top a^*$.
   - The performance

8.3 Algorithm and Regret

An algorithm is proposed for online linear optimization with delayed bandit feedback, achieving nearly optimal performance Ito et al. (2020) .

Algorithm 1 An algorithm for online linear optimization with delayed bandit feedback

Require: Action set $\mathcal{A}$, parameters T and $d \leq T-1$

Set $\gamma = 4\log(10mT)$ and $\eta = \min\left\{\sqrt{\frac{m\log T}{2(d+em)T}}, \frac{1}{100\gamma^2(2(d\sqrt{m}+m)}\right\}$.

Define $w_1 : \mathcal{B} \to \mathbb{R}_{>0}$ by $w_1(x) = 1$ for all $x \in B$.

for $t = 1, 2, \ldots, T$ **do**

  Let p_t be a distribution whose density function is proportional to w_t in (9).

  Pick $b_t \sim p'_t$, e.g., by iteratively sampling $b \sim p_t$ until $\|b\|^2_{S(p_t)^{-1}} \leq m\gamma^2$ holds.

  if $t > d$ **then**

    Get feedback of $\ell_{t-d}^\top a_{t-d}$, construct an unbiased estimator of ℓ_{t-d} as

$$\hat{\ell}_{t-d} = \ell_{t-d}^\top a_{t-d} S(p_{t-d})^{-1} b_{t-d},$$

  and update w_t by

$$w_{t+1}(x) = w_t(x)\exp\left(-\eta\hat{\ell}_{t-d}^\top x\right)$$

  end if

  if $t \leq d$ **then**

    Let $w_{t+1} = w_t$.

  end if

end for

Theorem 8.3.1. *For arbitrary loss sequences $\{\ell_t\}_{t=1}^T$, the regret for ALgorithm8.3 is bounded as*

$$\mathbb{E}[R_T] \leq \max\left\{\sqrt{8m(d+em)T\log T} + 3, Cm(d\sqrt{m}+m)\log^3(dmT)\right\}$$

where $\mathbb{E}[\cdot]$ means the expectation taken w.r.t the internal randomization of the algorithm and $C > 0 is a global constant$.

1.

Bibliography

Ito, S. et al. (2020). Delay and cooperation in nonstochastic linear bandits. *Advances in Neural Information Processing Systems* 33, 4872–4883.

Lovász, L. and S. Vempala (2007). The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms* 30.3, 307–358.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India* 2 (1), 49–55.