

## 1 Last Time

Last time, we covered a multiplicative weights algorithm for learning with small regret. Let  $\ell^t \in [-1, 1]^N$  be a “loss vector.” Define  $\Delta_N = \{x \in [0, 1]^N \mid \sum_{i=1}^N x_i = 1\}$  to be the  $N$ -dimensional probability simplex. We showed in the last lecture that:

**Theorem 16.1.** *For every  $0 < \varepsilon \leq 1$ , there exists an algorithm  $\text{Hedge}(\varepsilon)$  such that for all times  $T > 0$ , for every sequence of loss vectors  $(\ell^1, \dots, \ell^T)$ , and for every  $i \in \{1, \dots, n\}$ , at every time  $t \leq T$ ,  $\text{Hedge}(\varepsilon)$  produces  $p^t \in \Delta_N$  such that*

$$\sum_{t=1}^T \langle \ell^t, p^t \rangle \leq \sum_{t=1}^T \langle \ell^t, e_i \rangle + \varepsilon T + \frac{\ln N}{\varepsilon},$$

where  $e_i$  is the  $i$ th vector in the standard basis of  $\mathbb{R}^N$ . Note that the first term on the right hand side represents the loss of the  $i$ th expert, and the last two terms represents the regret of not having always chosen the  $i$ th expert.

Note that if we choose  $\varepsilon = \sqrt{\frac{\ln N}{T}}$ , then  $\varepsilon T + \frac{\ln N}{\varepsilon} = 2\sqrt{T \ln N}$ , so that the regret term is *sublinear* in time  $T$ . This indicates that the average regret of  $\text{Hedge}(\varepsilon)$  converges towards the best expert, so that  $\text{Hedge}(\varepsilon)$  is in some sense “learning”.

For future reference, we state the analogous result for gains  $g^t$  instead of losses  $\ell^t$ , i.e.,  $g^t = -\ell^t$ .

**Theorem 16.2.** *For every  $0 < \varepsilon \leq 1$ , there exists an algorithm  $\text{Hedge}_g(\varepsilon)$  such that for all times  $T > 0$ , for every sequence of gain vectors  $(g^1, \dots, g^T)$ , and for every  $i \in \{1, \dots, n\}$ , at every time  $t \leq T$ ,  $\text{Hedge}_g(\varepsilon)$  produces  $p^t \in \Delta_N$  such that*

$$\sum_{t=1}^T \langle g^t, p^t \rangle \geq \sum_{t=1}^T \langle g^t, e_i \rangle - \varepsilon T - \frac{\ln N}{\varepsilon},$$

where  $e_i$  is the  $i$ th vector in the standard basis of  $\mathbb{R}^N$ . Note that the first term on the right hand side represents the gain of the  $i$ th expert, and the last two terms represents the regret of not having always chosen the  $i$ th expert.

We also state a corollary of Theorem 16.2 that we will use.

**Corollary 16.3.** *Let  $\rho \geq 1$ . For every  $0 < \varepsilon \leq 1$ , for all times  $T \geq \frac{4\rho^2 \ln N}{\varepsilon^2}$ , for all sequences of gain vectors  $(g^1, \dots, g^T)$  with each  $g^t \in [-\rho, \rho]^N$ , and for all  $i \in \{1, \dots, N\}$ , at every time  $t \leq T$ ,  $\text{Hedge}_g(\varepsilon)$  produces  $p^t \in \Delta_N$  such that*

$$\frac{1}{T} \sum_{t=1}^T \langle g^t, p^t \rangle \geq \frac{1}{T} \sum_{t=1}^T \langle g^t, e_i \rangle - 2\varepsilon.$$

*Proof.* Multiply both the left and right sides of Theorem 16.2 by  $1/T$ , and observe that  $\frac{\ln N}{\varepsilon T} \leq \frac{\varepsilon}{4\rho^2} \leq \varepsilon$  when  $T \geq \frac{4\rho^2 \ln N}{\varepsilon^2}$  and  $\rho \geq 1$ . As a result,

$$\frac{1}{T} \sum_{t=1}^T \langle g^t, p^t \rangle \geq \frac{1}{T} \sum_{t=1}^T \langle g^t, e_i \rangle - \varepsilon - \frac{\ln N}{\varepsilon T} \geq \frac{1}{T} \sum_{t=1}^T \langle g^t, e_i \rangle - 2\varepsilon.$$

□

## 2 Zero-Sum Games

A zero sum game can be described by a pay-off matrix  $M \in \mathbb{R}^{m \times n}$ , where the two players are the “row player” and the “column player”. Simultaneously, the row player chooses a row  $i$  and the column player chooses a column  $j$ , and the row player receives a pay-off of  $M_{i,j}$ . Alternatively, the column player loses  $M_{i,j}$ ; hence, the name “zero-sum”.

Given a strategy  $p \in \Delta_m$  for the row player and a strategy  $q \in \Delta_n$  for the column player, the expected pay-off to the row player is

$$\mathbb{E}[\text{pay-off to row}] = p^\top M q = \sum_{i,j} p_i q_j M_{i,j}.$$

The row player wants to maximize this value, while the column player wants to minimize this value. Note that the choices of  $x$  and  $y$  are made simultaneously. Later, we will see that these choices really do not have to be made at the same time.

Suppose row player fixes a strategy  $p \in \Delta_m$ . Knowing the row player’s strategy, the column player can choose a response to minimize the row player’s expected winnings:

$$C(p) = \min_{q \in \Delta_n} p^\top M q = \min_{j \in [n]} p^\top M e_j.$$

The equality holds because if the column player already knows the row player’s strategy ( $p$ ), then the column player’s best strategy is to choose the column that minimizes the row player’s expected winnings.

Analogously, suppose the column player fixes a strategy  $y \in \Delta_n$ . Knowing the column player’s strategy, the row player can choose a response to maximize his own expected winnings:

$$R(q) = \max_{p \in \Delta_m} p^\top M q = \max_{i \in [m]} e_i^\top M q.$$

Overall, the row player wants to achieve  $\max_{p \in \Delta_m} C(p)$ , and the column player wants to achieve  $\min_{q \in \Delta_n} R(q)$ . It is easy to see that for any  $p \in \Delta_m, q \in \Delta_n$ , we have

$$C(p) \leq R(q) \tag{16.1}$$

Intuitively, in the latter the column player commits to a strategy  $q$ , and hence gives more power to the row player. Formally, the row player could always play strategy  $p$  in response to  $q$ , and hence could always get value  $C(p)$ . But  $R(q)$  is the best response, which could be even higher. In fact, there exists some strategies  $p \in \Delta_m, q \in \Delta_n$  such that:

$$C(p) = R(q) \tag{16.2}$$

This is formalized by the following theorem:

**Theorem 16.4.** (Von Neumann's Minimax) For any finite zero-sum game  $M \in \mathbb{R}^{m \times n}$ ,

$$\max_{p \in \Delta_m} C(p) = \min_{q \in \Delta_n} R(q).$$

This common value  $V$  is called the value of the game  $M$ .

*Proof.* By scaling, we assume for the sake of contradiction that  $\exists M \in [-1, 1]^{m \times n}$  such that  $\max_{p \in \Delta_m} C(p) < \min_{q \in \Delta_n} R(q) - \delta$  for  $\delta > 0$ .

We treat each row of  $M$  as an expert. At each time step  $t$ , the row player produces  $p^t \in \Delta_m$ . Initially,  $p^1 = (\frac{1}{m}, \dots, \frac{1}{m})$ , which represents that the row will choose any row with equal probability when he has no information to work with.

At each time  $t$ , the column player plays the best response to  $p^t$ , i.e.,

$$j_t := \arg \max_{j \in [n]} (p^t)^\top M e_j.$$

Now define the gain vector for the row player as

$$g_t := M e_{j_t}.$$

which is the  $j$ -th column of  $M$ . This the row player uses to update the weights as get  $p_{t+1}$ , etc. Define

$$\hat{p} := \frac{1}{T} \sum_{t=1}^T p^t \quad \text{and} \quad \hat{q} := \frac{1}{T} \sum_{t=1}^T e_{j_t}.$$

These are the average long-term plays of the row player, and the best responses of the column player to those plays. We know that  $C(\hat{p}) \leq R(\hat{q})$  by (16.2). Now by Corollary 16.3, after  $T \geq \frac{4 \ln m}{\varepsilon^2}$  steps,

$$\begin{aligned} \frac{1}{T} \sum_t \langle p^t, g^t \rangle &\geq \max_i \frac{1}{T} \sum_t \langle e_i, g^t \rangle - 2\varepsilon && \text{(by Hedge)} \\ &= \max_i \left\langle e_i, \frac{1}{T} \sum_t g^t \right\rangle - 2\varepsilon \\ &= \max_i \left\langle e_i, M \left( \frac{1}{T} \sum_t e_{j_t} \right) \right\rangle - 2\varepsilon && \text{(by definition of } g_t) \\ &= \max_i \langle e_i, M \hat{q} \rangle - 2\varepsilon \\ &= R(\hat{q}) - 2\varepsilon \end{aligned}$$

We also know that since  $p^t$  is the row player's strategy, and  $C$  is concave (i.e., the payoff on the average strategy  $\hat{p}$  is no more than the average of the payoffs)<sup>1</sup>,

$$\frac{1}{T} \sum \langle p^t, g^t \rangle = \frac{1}{T} \sum C(p^t) \leq C \left( \frac{1}{T} \sum p^t \right) = C(\hat{p})$$

Putting it all together:

$$R(\hat{q}) - 2\varepsilon \leq C(\hat{p}) \leq R(\hat{q})$$

Since we control  $\varepsilon$ , for any  $\delta$  we can make  $\varepsilon$  smaller than  $\delta/2$  to get the contradiction.  $\square$

<sup>1</sup>To see this, recall that  $C(p) = \min_q p^\top M q$ . Let  $q^*$  be the optimal value of  $q$  that minimizes  $C(p)$ . Then for any  $a, b \in \Delta_m$ , we have that  $C(a + b) = (a + b)^\top M q^* = a^\top M q^* + b^\top M q^* \geq \min_q a^\top M q + \min_q b^\top M q = C(a) + C(b)$ .

### 3 Solving LPs

In this lecture, we use the above theorem to solve LPs approximately. We are given an LP with  $n$  variables and  $m$  constraints (excluding non-negativity constraints):

$$\begin{aligned} \max \quad & c \cdot x \\ & Ax \leq b \\ & x \geq 0 \end{aligned}$$

#### 3.1 The Oracle

First suppose there is only a single constraint. I.e., suppose  $\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}$ , and we want to solve

$$\begin{aligned} \max \quad & c \cdot x \\ & \alpha \cdot x \leq \beta \\ & x \geq 0 \end{aligned}$$

We make a simple observation.

**Proposition 16.5.** *Given a single constraint  $\alpha \cdot x \leq \beta$ , there is a fast oracle that finds  $x \in K$  such that  $\alpha \cdot x \leq \beta$  if such an  $x$  exists, or else outputs that no such  $x$  exists.*

*Proof.* We give the proof only for the case where  $c_i \geq 0$  for all  $i$ ; the general case is left as an exercise. Denote by  $OPT$  the optimal value of the LP. Find  $j^* = \operatorname{argmax}_j \frac{\alpha_j OPT}{c_j}$ . Set  $x^* = \frac{OPT}{c_{j^*}} e_{j^*}$ . Then  $x^*$  is a vector that maximizes  $\alpha \cdot x$  subject to  $x \in K$ . Now check whether  $\alpha \cdot x^* \leq \beta$  and output  $x^*$  if the inequality is satisfied; otherwise output no such  $x$  exists.

Note that the runtime of this oracle procedure is polynomial in  $n$ , the dimension of the problem. (As we'll see later in the lecture, when we consider max-flows, that even if the dimension is exponential, we may be able to efficiently figure out the  $\operatorname{argmax}$  in the calculation above.)  $\square$

Therefore, solving an LP with 1 constraint is easy. What about more general LPs with multiple constraints? The key idea is to combine the multiple constraints using weights. We will call the above oracle while using Hedge by repeatedly asking the oracle to produce a solution to a convex combination of the  $m$  constraints and updating the weight/importance of each constraint based on how badly the constraint was violated by the current solution. To this end, we will set our gain vectors  $p^t \in \Delta_m$  to be the amount of *violation* of the corresponding constraint. The intuition is simple: greater violation means more gain and hence more weight in the next iteration, which forces us to not violate it as much.

An upper bound on the maximum possible violation is the *width*  $\rho$  of the LP, defined by

$$\rho := \max_{x \in K, i \in [m]} \{ |a_i \cdot x - b_i| \}. \quad (16.3)$$

We will assume that  $\rho \geq 1$ .

#### 3.2 The Algorithm

Below we give the formal algorithm to approximately solve the LP.

1.  $p^1 \leftarrow (1/m, \dots, 1/m)$

2. Define  $\alpha^t := \sum_{i=1}^m p_i^t a_i$ , and  $\beta^t = \sum_{i=1}^m p_i^t b_i$ . Call the oracle to find  $x^t$  such that  $\alpha^t, x^t \leq \beta^t$  (i.e.,  $\sum_{i=1}^m \langle p_i^t a_i, x^t \rangle \leq \sum_{i=1}^m p_i^t b_i$ ). If oracle says infeasible, it means that the original LP is infeasible, as any feasible solution to the original LP would satisfy a convex combination of the LP's constraints.
3. Define gain vector by  $g_i^t := \langle a_i, x^t \rangle - b_i$ .
4. Update  $p^{t+1}$  using Hedge( $\varepsilon$ ).
5. Run steps 1 to 4 for  $T := \Theta(\rho^2 \ln m / \varepsilon^2)$  rounds.
6. return  $\hat{x} \leftarrow (x^1 + \dots + x^T) / T$ .

### 3.3 The Analysis

**Theorem 16.6.** *For every  $0 \leq \varepsilon \leq 1/4$ , the above algorithm returns  $\hat{x} \in K$  such that*

1.  $c^\top \hat{x} \geq OPT$ ,
2.  $\langle a_i, \hat{x} \rangle - b_i \leq 2\varepsilon$  for all  $i \in [m]$ .

Moreover, the algorithm calls the oracle at most  $O(\rho^2 \ln m / \varepsilon^2)$  times.

*Proof.* For the first part, observe that if the original LP was feasible then the same solution is feasible for the new problem. To show this, let the solution in the original problem be  $x^*$ . For any  $p^t$ , we have  $p^t A x^* \leq p^t b = \beta^t$  and  $c^\top x^* = OPT$ . At each iteration,  $x^t$  satisfies  $c^\top x^t \leq OPT$  which implies that  $\hat{x}$  also satisfies  $c^\top \hat{x} \leq OPT$  (since  $\hat{x}$  to be the average of  $x^t$ 's).

For the second part, let  $i \in [m]$ . Define  $\alpha^t = \sum_{i=1}^m p_i^t a_i$  and  $\beta^t = \sum_{i=1}^m p_i^t b_i$ . Then

$$\begin{aligned}
 \langle p^t, g^t \rangle &= \langle p^t, A x^t - b \rangle \\
 &= \langle p^t, A x^t \rangle - \langle p^t, b \rangle \\
 &= \langle \alpha^t, x^t \rangle - \beta^t \\
 &\leq 0.
 \end{aligned}$$

So the left hand side in Corollary 16.3 satisfies:

$$\frac{1}{T} \sum_{t=1}^T \langle p^t, g^t \rangle \leq 0.$$

Second, the right hand side in Corollary 16.3 satisfies:

$$\begin{aligned}
 \frac{1}{T} \sum_{t=1}^T \langle p^t, g^t \rangle &\geq \max_i \frac{1}{T} \sum_{t=1}^T \langle e_i, g^t \rangle - 2\varepsilon \\
 &= \max_i \langle e_i, \frac{1}{T} \sum_{t=1}^T g^t \rangle - 2\varepsilon \\
 &= \max_i \frac{1}{T} \sum_{t=1}^T (\langle a_i, \hat{x}^t \rangle - b_i) - 2\varepsilon \\
 &= \max_i (\langle a_i, \hat{x} \rangle - b_i) - 2\varepsilon
 \end{aligned}$$

By Corollary 16.3, we have

$$0 \geq \frac{1}{T} \sum_{t=1}^T \langle p^t, g^t \rangle \geq \max_i (\langle a_i, \hat{x} \rangle - b_i) - 2\varepsilon$$

Hence, for each constraint  $i$ , we have that  $\langle a_i, \hat{x} \rangle - b_i - 2\varepsilon \leq 0$  so  $\langle a_i, \hat{x} \rangle - b_i \leq 2\varepsilon$ . This implies that each solution is  $2\varepsilon$ -feasible as claimed.  $\square$

### 3.4 A Small Extension: Approximate Oracles

Recall the definition of the problem width from (16.3). Two comments:

- In the above analysis, we do not care about the maximum value of  $|a_i \cdot x - b_i|$  over all points  $x \in K$ , we only care about the largest this expression can get over all the points that can be potentially returned by the oracle. While this seems a pedantic point, it will be useful — if there are many solutions we can return for  $\alpha \cdot x \leq \beta$ , we can try to return the one with least width. But we can do more, as the next point outlines.
- We can also relax the oracle to satisfy  $\alpha \cdot x \leq \beta + \delta$  for some small  $\delta > 0$  instead. Define the *width* of the LP with respect the relaxed oracle to be

$$\rho_{\text{relaxed}} := \max_{i \in [m], x \text{ returned by relaxed oracle}} \{|a_i \cdot x - b_i|\}. \quad (16.4)$$

The new range of  $g^t$  using the relaxed oracle will be  $[-\rho_{\text{relaxed}}, \rho_{\text{relaxed}}]^N$ . So running the same algorithm with the relaxed oracle will give us that  $a_i \cdot \hat{x} \leq b_i + \varepsilon + \delta$  for all  $i \in [m]$  and the number of calls to the relaxed oracle is  $O(\rho_{\text{relaxed}}^2 \ln m / \varepsilon^2)$ .

We will not use these today, but these will be crucial to us in the next lecture.

### Acknowledgments

These lecture notes were scribed by Paul Liang, based on previous scribe notes of Dongho Choi, Yang Jiao and Edgar Chen.