Online Algorithms

Basic setting

- input : $\sigma = (\sigma_1, \dots, \sigma_n)$

option 1: n is given in the beginning.
option 2: σ_n additionally indicates that there's no more input.

- on day i,
  * σ_i is revealed.
  * algorithm makes some "irrevocable decision" π_i.
    (only based on previous information, so formally, algorithm on day i is just a function mapping $(\sigma_1, \pi_1, \dots, \sigma_i)$ to π_i.)
  * of course, π_i needs to satisfy some constraints depending on $(\sigma_1, \pi_1, \dots, \sigma_i)$.

- after day n,
  * output $\pi = (\pi_1, \dots, \pi_n)$ is determined.
  * Let $val(\pi) \in \mathbb{R}_{\geq 0}$ be the quality (or cost) of π.
  * Let $OPT(\sigma) = \pi^* = (\pi_1^*, \dots, \pi_n^*)$ be the optimal solution s.t.
    ① $\forall i \in [n]$, π_i^* is a valid solution given $\sigma_1, \pi_1^*, \dots, \sigma_i$.
    ② $val(\pi^*)$ is optimal.
    (note that π^* depends on entire σ, which makes it "offline optimal")

- Competitive ratio $= \begin{cases} \max_\sigma \dfrac{val(ALG(\sigma))}{val(OPT(\sigma))} & \text{(for minimization problem)} \\ \min_\sigma \dfrac{val(OPT(\sigma))}{val(ALG(\sigma))} & \text{(for maximization problem)} \end{cases}$

Ski Rental

Setting

- You are on a ski trip (of unknown length).
- On day $i=1,\dots,$ (Let's assume $\sigma_i = Y$)
  * input σ_i : Y (trip continues on day i), N (trip is done).

  (The possible inputs are $I_1,\dots,I_k,\dots,$ where $I_k = (\overbrace{Y,Y\dots,Y}^{k},N)$)
  * output: you can choose to either (B∈ℕ is known in the beginning)

    (i) rent: \$1, only for that day.

    (ii) buy: \$B, so you don't worry in the future.

  Then, the algorithms are $Alg_1,\dots,Alg_k,\dots,$ where
  $Alg_k \equiv$ rent on day $1\dots k-1$ and buy on day k.
- Then, if input is I_j and algo is Alg_i, then

$$val(OPT(I_j)) = \min(j, B).$$

$$val(ALG_i(I_j)) = \begin{cases} (i-1+B) & \text{if } i \le j. \\ j & \text{if } j < i. \end{cases}$$

Given B, which ALG_i should we use in order to min. competitive ratio?

Claim. Competitive ratio of ALG_B is $\le 2 - 1/B$.

Pf. If $j < B$, $\quad OPT(I_j) = j$ — let's write $OPT(I)$ for $val(OPT(I))$.

$ALG_B(I_j) = j$.

If $j \ge B$ $\quad OPT(I_j) = B$

$ALG_B(I_j) = 2B - 1$.

So, for any $j \in \mathbb{N}$, $ALG(I_j)/OPT(I_j) \le 2 - 1/B$ $\square$.

Claim. For any $i \in \mathbb{N}$, Alg_i is $\ge 2 - 1/B$ competitive.

Pf. $Alg_i(I_i) = (i-1) + B$. and $OPT(I_i) = \min(i, B)$, so

$Alg_i(I_i)/OPT(I_i) \ge 2 - 1/B$ $\square$

Randomized Online Algorithms

Basic setting

option 1: n is given in the beginning.
option 2: σ_n additionally indicates that there's no more input.

- input: $\sigma = (\sigma_1, \dots, \sigma_n)$
- on day i,
  * σ_i is revealed.
  * algorithm makes some randomized "irrevocable decision" π_i.
    (only based on previous information, so formally, algorithm on day i is just a randomized function mapping $(\sigma_1, \pi_1, \dots, \sigma_i)$ to π_i.)
  * of course, π_i needs to satisfy some constraints depending on $(\sigma_1, \pi_1, \dots, \sigma_i)$.
- after day n,
  * output $\pi = (\pi_1 \dots \pi_n)$ is determined.
  * Let $val(\pi) \in \mathbb{R}_{\geq 0}$ be the quality (or cost) of π.
  * Let $OPT(\sigma) = \pi^* = (\pi_1^*, \dots, \pi_n^*)$ be the optimal solution s.t.
    ① $\forall i \in [n]$, π_i^* is a valid solution given $\sigma_1, \pi_1^*, \dots, \sigma_i$.
    ② $val(\pi^*)$ is optimal.
    (note that π^* depends on entire σ, which makes it "offline optimal")
- Competitive ratio $= \begin{cases} \max_\sigma \dfrac{\mathbb{E}[val(ALG(\sigma))]}{val(OPT(\sigma))} & \text{(for minimization problem)} \\ \min_\sigma \dfrac{val(OPT(\sigma))}{\mathbb{E}[val(ALG(\sigma))]} & \text{(for maximization problem)} \end{cases}$

Note σ is fixed before $\mathbb{E}$! (Adversary cannot change σ after seeing our random decision) — "oblivious adversary" model.

Randomized Ski Rental

Then our "randomized algo" is a probability distribution $p_1, \dots, p_k, \dots$ where $p_i = \Pr[\text{we play } Alg_i]$

Our goal: $\forall I_j$, $\dfrac{\mathbb{E}_i[Alg_i(I_j)]}{OPT(I_j)} = \dfrac{\sum_i p_i \cdot Alg_i(I_j)}{OPT(I_j)} \le c$ for some small $c \in (1,2)$

Assume $B = 4$.

	I_1	I_2	I_3	I_4	I_5	...
Alg_1	4, 1	4, 2	4, 3	4, 4	4, 4	
Alg_2	1, 1	5, 2	5, 3	5, 4	5, 4	
Alg_3	1, 1	2, 2	6, 3	6, 4	6, 4	
Alg_4	1, 1	2, 2	3, 3	7, 4	7, 4	
⋮						

ALG

OPT

A: their matrix. (blue values)

Our goal: compute $p = (p_1 \dots p_k \dots)$ s.t. $(p^T A)_j \le c \cdot j$. $\forall j \in \mathbb{N}$.

$\Rightarrow$ exactly zero-sum game.

Suppose we only use $p_1 \dots p_4$ (i.e., $p_i = 0 \ \forall i \ge 5$).

Then $I_4, I_5, \dots$ are all same (for both ALG / OPT), so we only need to consider I_1, I_2, I_3, I_4.

Then, Our goal becomes solve following LP.

minimize c

subject to $p_1+p_2+p_3+p_4=1$ (E0)

$4p_1+p_2+p_3+p_4 \le c$ (E1)

$4p_1+5p_2+2p_3+2p_4 \le 2c$ (E2)

$4p_1+5p_2+6p_3+3p_4 \le 3c$ (E3)

$4p_1+5p_2+6p_3+7p_4 \le 4c.$ (E4)

$p \ge 0$

If we assume $(E_1), (E_2), (E_3), (E_4)$ hold with equality, (WITH loss of generality)

$(E4)-(E3)$: $4p_4 = c$

$(E3)-(E2)$: $4p_3+p_4 = c$

$(E2)-(E1)$: $4p_2+p_3+p_4 = c$

$(E1)-(E0)$: $3p_1 = c-1.$

$\Rightarrow p_4 = c/4.$ $p_3 = ((3/4)c)/4 = (3/16)c.$ $p_2 = (1-1/4-3/16)c/4 = (9/64)c.$

$p_1 = (c-1)/3.$

$c = 1/(1-(1-1/4)^4) = 1/(1-81/256) = 256/175 \approx 1.46$ satisfies (E0)-(E4).

(for general B, $c = 1/(1-(1-1/B)^B) \le e/(e-1) \fallingdotseq 1.588$ is the optimal value.)