Lecture 03

classic online algorithms and analysis PART II

(cont. then randomized algorithms)

- one-way trading

Description : sell all stocks in a single day w/ n trading time slots.

Input: $(P_1, \dots, P_n)$ rate/price at time $i=1,\dots,n$

$L \le P_i \le U$, $\forall i$ → bounds are known in advance.

Output: $f_1, \dots, f_n \in [0,1]$

$f_i :=$ fraction of the total stocks traded at time i

$\sum_{i=1}^{n} f_i = n$

objective: $\max_{\{f_i\}} \sum_{i=1}^{n} f_i P_i$ fixed

s.t. $\sum_{i=1}^{n} f_i = 1$, $f_i \ge 0$, $\forall i$

$(L \le P_i \le U , \forall i)$

- We have to trade stocks online. i.e., decide f_i at time i without seeing future prices $P_{i+1}, \dots, P_n$.

Q: How to design competitive online algorithms ?

Before this, let's consider a simpler problem :

- Time-series search

Description: A discrete version of one-way trading

Input: the same

output: index $i \in [n]$

objective : Compute an index $i \in [n]$ to maximize P_i. i.e.

$$\max_{i \in [n]} P_i$$

s.t. $L \le P_i \le U$, $\forall i$

Remark :

- Q: Which problem is harder for online algorithms ?

  To be more specific. to achieve a lower competitive ratio. (for maximization problems. we reverse the numerator / denominator $\frac{OPT}{ALG}$ → abuse notation ! ignore $c(\cdot)$ or $r(\cdot)$ for simplicity.

- Indeed, the second (discrete) version is harder.

  It can motivate an online algorithm design for the 1st problem.

  Consider the following :

- Let $\phi := \frac{U}{L}$.

  Any algorithm can achieve this. Can we do better ?

Algorithm The reservation price policy (RPP)

procedure reservation price

$p^* \leftarrow \sqrt{UL}$

flag $\leftarrow 0$ # already traded or not

$j \leftarrow 1$

while $j \leq n$, flag $= 0$ do

  if $j < n$ & $p_j \geq p^*$, then

    Trade all stocks at time j

    flag $\leftarrow 1$

  else if $j = n$, then

    Trade all stocks at time n.

- This is a threshold - based policy.

Theorem 3.1

The reservation price algorithm achieves a competitive ratio $\sqrt{\phi}$, even w/o knowing n.

proof: • when $p_j < p^*$, $\forall j \in [n]$.

RPP gets $p_n \geq L$.

OPT gets $\max_j p_j \leq p^* = \sqrt{UL}$

$$\Rightarrow \frac{OPT}{RPP} \leq \frac{\sqrt{UL}}{L} = \sqrt{\phi}$$

• when $p_j \geq p^*$ for some $j \in [n]$

Denote by i the smallest such an index

RPP get $p_i \geq p^* = \sqrt{UL}$

OPT get $\max_j p_j \leq U$

$$\Rightarrow \frac{OPT}{RPP} \leq \frac{U}{\sqrt{UL}} = \sqrt{\phi} \quad \#$$

• Is this optimal?

Yes!

Theorem 3.2

There is no deterministic algorithm that can achieve a competitive ratio smaller than $\sqrt{\phi}$, even n is known.

proof: consider the following construction:

$$\underbrace{\sqrt{UL}, \sqrt{UL}, \cdots, \sqrt{UL}}_{n-1}, \begin{cases} \text{if ALG trades at } i \leq n-1: \\ p_n = U \\ \text{if ALG trades at } i = n: \\ p_n = L. \end{cases}$$

we see that

$$CR(ALG) \geq \min\left\{ \frac{U}{\sqrt{UL}}, \frac{\sqrt{UL}}{L} \right\} = \phi, \quad \forall ALG. \qquad \#$$

- Q: Is knowing U and L necessary?

  Yes.

Theorem 3.3

Suppose instead of U, L, only ϕ is known a priori. Then any alg cannot have a competitive ratio better than ϕ.

proof: consider

$$\underbrace{1, 1, \cdots, 1}_{n-1} \begin{cases} \text{if ALG trades at } \leq n-1: \\ \quad P_n = \phi \\ \text{if ALG trades at } n: \\ \quad P_n = 1/\phi \end{cases}$$

Similar argument yields a lower bound ϕ on $CR(ALG)$. #

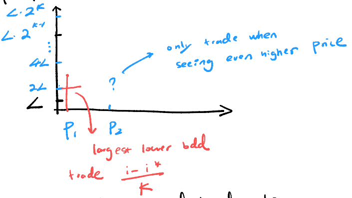
- Can we have an extension of RPP for the one-way trading problem? WLOG, assume $\phi = 2^k$ for $k \in \mathbb{N}_+$.

Algorithm The Mixture of RPPs (MRPP)

procedure Reservation Price

```
i* ← -1
for j ← 1 to n do
    i ← max{ i | L2^i ≤ P_j }
    if i = k, then
        i ← k - 1
    if i > i*, then
        trade fraction (i - i*)/k =: f_i of stocks at j
        i* ← i
Trade all remaining stocks at time n.
```

- Intuition behind:
  - Exponentially spaced reservation prices
    $L \cdot 2^i, \; i \in \{0, 1, \dots, k\}$
  - Index i^* tracks the best reservation price that has been exceeded



- A generalized threshold-based alg.

Theorem 3.4.

The competitive ratio of MRPP is at least

$$\log_2 \phi \cdot \left(\frac{1}{1 - \frac{1}{\phi \ln 2 \cdot 2^{\frac{1}{\ln 2}} - 1}} \right).$$

proof: Fix any input prices $p_1, \dots, p_n$. Let $i_1, \dots, i_n$ be the corresponding reservation indices, i.e.,

$$i_j := \max\{ j \mid L 2^i \leq p_j \}.$$

Denote $\ell \in \arg\max_{j \in [n]} p_j$ (arbitrarily choose one to break the tie). So, $OPT = p_\ell \leq L 2^{i_\ell + 1}$.

Let $i_0^* < \dots < i_m^*$ $(m \geq 1)$ be the non-decreasing sequence of tracking indices. By definition, $i_0^* = -1$ and $i_m^* = i_\ell$.

Now, let's compute the total gain of MRPP, which is at least

$$\underbrace{\sum_{j=1}^{m} \frac{i_j^* - i_{j-1}^*}{k} L 2^{i_j^*}}_{\text{first } m \text{ sells.}} + \underbrace{\frac{k - i_\ell}{k} \cdot L}_{\text{remaining stocks sold at time } n \text{ price} \geq L} \quad \text{——} (*)$$

(1) (2)

Worst case prices that minimize (1) should reduce the gap $i_j^* - i_{j-1}^*$, $\forall j$. Hence, it induces $i_j^* = j-1$ (no skip selling).

yielding

$$(1) = \sum_{j=1}^{i_\ell} 2^j \cdot L = (2^{i_\ell+1} - 1) L.$$

Combining above w/ $(*)$,

$$ALG \geq \frac{2^{i_\ell+1} - 1}{k} L + \frac{k - i_\ell}{k} \cdot L$$

Thus,

$$\frac{OPT}{ALG} \leq \frac{L 2^{i_\ell+1}}{\frac{2^{i_\ell+1} - 1}{k} L + \frac{k - i_\ell}{k} \cdot L}$$

$$= k \frac{2^{i_\ell+1}}{2^{i_\ell+1} + k - i_\ell - 1}$$

Minimizing over i_ℓ.

$$\frac{OPT}{ALG} \leq k \cdot \left(\frac{2^{k-1+\frac{1}{\ln 2}}}{2^{k-1+\frac{1}{\ln 2}} - \frac{1}{\ln 2}} \right)$$

when $i_\ell = k-1 + \frac{1}{\ln 2}$.

$$\leq \log \phi \left(\frac{1}{1 - \frac{1}{\phi \ln 2 \cdot 2^{\frac{1}{\ln 2} - 1}}} \right)$$

#

Remark. HW

- Can generalize Thm 3.3 to one-way trading as well.
- what is the best one can do?

  Is it possible to achieve a better CR?

Yes!

Theorem 3.5

Fix L, and U. Any alg cannot have a CR smaller than

$$\ln\left(\frac{U}{L}\right) + 1.$$

Furthermore. $\exists$ algs s.t. $\ln\left(\frac{U}{L}\right)+1$ is achievable.

- threat-based alg El-Yaniv, Karp. Turpin, 2001
- CR-Pursuit Lin et. al. 2019.
- primal-dual analysis.

Algorithm. $\phi(x): [0,1] \to [0, \infty)$

Input : $\phi(x)$. $x^{(0)}=0$. $p_1 \cdots p_n$

for $j \leftarrow 1$ to n. do

$$f_i \leftarrow \underset{f \in \mathbb{R}_+}{\arg\max} \ \underbrace{p_i \cdot f - \int_{x^{(i-1)}}^{x^{(i-1)}+f} \phi(\delta)\, d\delta}_{\text{psedo-revenue}}$$

$x^{(i)} \leftarrow x^{(i-1)} + f_i$

If $x^{(i)} > 1$: $f_i = 0$

Theorem. 3.6

The presented alg is $(\ln(\phi)+1)$- competitive.

proof idea:

construct a ϕ and use the following primal-dual analysis:

primal problem: $\max \sum_{i=1}^{n} p_i \overset{\text{fixed}}{} f_i$

(P)

$$\text{s.t. } \sum_{i=1}^{n} f_i \le 1$$

$$f_i \ge 0. \quad \forall i \in [n]$$

Dual problem:

(D) min λ

$$\text{s.t. } \lambda \ge p_i \quad \forall i \in [n].$$

Lemma 3.1 Denote by P_i. D_i the objective values of (P) & (D) respectively after processing p_i.

An online alg for one-way trading is α-competitive if it can determine primal variables x and construct dual variables λ based on the primal variables s.t.

(Feasible sol) x. λ are feasible;

(Initial Ineq) $\exists$ an index $k \in [n] \cup \{0\}$ s.t.

$$P_k \ge \frac{1}{\alpha} D_k$$

(Incremental Ineq) $\forall i \in \{k+1, \dots, n\}$.

$$P_i - P_{i-1} \ge \frac{1}{\alpha}(D_i - D_{i-1}).$$

Proof: one-line proof:

$$ALG = P_n \geq \frac{1}{\alpha} D_n \underset{(1)}{\geq} \frac{1}{\alpha} D^* \underset{(2)}{\geq} \frac{1}{\alpha} OPT$$

assumption

(1) dual problem property

(2) weak duality

$$P_n - P_K \geq \frac{1}{\alpha}(P_n - P_K)$$

since $P_K \geq \frac{1}{\alpha} D_K$, $\Rightarrow P_n \geq \frac{1}{\alpha} D_n$. #

HW: use this lemma to prove the following theorem and use the Gronwall's inequality to find the best α.

Theorem 3.7.

The presented alg is α-competitive if ϕ is

$$\phi(x) = \begin{cases} L, & x \in [0, \beta] \\ \varphi(x), & x \in (\beta, 1] \end{cases}$$

w/ a level parameter $\beta \in [\frac{1}{\alpha}, 1]$, and a differentiable φ s.t. satisfies

$$\begin{cases} \varphi(x) \geq \frac{1}{\alpha}\varphi'(x), & x \in [\beta, 1], \\ \varphi(\beta) = L, \ \varphi(1) \geq U. \end{cases}$$

∎

Lecture 04 Randomized Online Algorithms PART I

- online knapsack

Description: Pack maximum total weight of items into a single knapsack

c.f. multi-knapsack

item weight

Input: $(w_1, \dots, w_n)$, $w_i \in \mathbb{R}_+$, $\forall i \in [n]$

$W > 0$. knapsack capacity. Known in advance.

Output: $z = (z_1, \dots, z_n)$, $z_i \in \{0, 1\}$, $\forall i \in [n]$

objective:

$$\max \sum_{i=1}^{n} z_i w_i$$

$$\text{s.t.} \sum_{i=1}^{n} z_i w_i \le W$$

Remarks

- Can you find a deterministic online algorithm that achieves a constant competitive ratio?

In general, no.

(Given bounds on w_i, similar to the bounds L, U on P_i for one-way trading, then yes, and can be optimal!)

Theorem 4.1

Let $\varepsilon > 0$ be arbitrary. Then any deterministic online algorithm ALG for the online knapsack problem must satisfy:

$$CR(ALG) \geq \frac{1-\varepsilon}{\varepsilon}.$$

[Strong] can choose parameters $\{w_1 \cdots w_n\}$ after seeing the online strategy!

Proof. Idea: construct an adversary.

Let $n \in \mathbb{N}_+$, WLOG, assume $W = n$.

$$\underbrace{\varepsilon n, \varepsilon n, \cdots \varepsilon n}_{i}, \underbrace{n(1-\varepsilon)+\varepsilon}_{1}, \underbrace{0, \dots, 0}_{n-i-1}$$

arbitrarily small

ALG decides to pack

If $i=n$, the ALG never packs so $ALG=0$, but $OPT \geq \varepsilon n$

$$\Rightarrow \frac{OPT}{ALG} = \infty.$$

If $i<n$, then ALG cannot pack the i-th item, since

$$\varepsilon n + (n-\varepsilon n)+\varepsilon > n = W$$

$$\Rightarrow ALG = \varepsilon n.$$

$$OPT = (n-\varepsilon n)+\varepsilon$$

$$\Rightarrow \frac{OPT}{ALG} = \frac{n(1-\varepsilon)+\varepsilon}{\varepsilon n} \geq \frac{1-\varepsilon}{\varepsilon}. \quad \#$$

- If w_i has a "lower bdd," $w_i \geq L > 0$, then this construction fails.
- Q: Without assuming bdded $\{w_i\}$, can you design a better online algorithm?

<u>Algorithm</u> Simple Randomized algorithm for online knapsack

procedure SimpleRandom

Sample a Bernoulli$(\frac{1}{2})$ random bit, $B \in \{0,1\}$

If $B=0$, then

Pack items $w_1 \ldots w_n$ greedily, i.e., pack the items sequentially whenever they fit in the knapsack.

Else

Pack the first item of weight $\geq \frac{W}{2}$ if $\exists$ such an item. Ignore the rest.

<u>Theorem</u> 4.1 [Randomization helps]

$$CR(\text{SimpleRandom}) \leq 4.$$

adversarially chosen

i.e. $$\frac{OPT}{\mathbb{E}[\text{SimpleRandom}]} \leq 4, \quad \forall \{w_1, \ldots, w_n\},\ n \geq 1.$$

$W \rightarrow$ known to ALG

Proof:

Case I: $\forall i \in [n],\ w_i < \frac{W}{2}$. (knapsack capacity) We consider the first mode ($B=0$)

subcases $\begin{cases} \sum_{i=1}^{n} w_i \leq W & \text{— (a)} \\ \sum_{i=1}^{n} w_i > W & \text{— (b)} \end{cases}$

For (a), ALG packs all items w/ probability $\frac{1}{2}$ ($B=0$).
(ALG = Simple Random)

$$\Rightarrow \mathbb{E}[ALG] \geq \frac{1}{2} \cdot \sum_{i=1}^{n} w_i$$

$$OPT = \sum_{i=1}^{n} w_i$$

$$\Rightarrow \frac{OPT}{\mathbb{E}[ALG]} \leq 2.$$

For (b). ALG will ignore some items. Let j be the first item that is not packed. ($B=0$)

We have:

$$\text{remaining capacity} \leq \underbrace{w_j < \frac{1}{2}W}_{\text{case I assumption}}$$

$$\Rightarrow \text{packed total weight} \geq W - \frac{1}{2}W = \frac{1}{2}W$$

$$\Rightarrow \mathbb{E}[ALG] \geq \frac{1}{2} \cdot \frac{1}{2}W = \frac{1}{4}W$$

$OPT \leq W$, so,

$$\Rightarrow \frac{OPT}{\mathbb{E}[ALG]} \leq 4.$$

Case II. $\exists\ i \in [n]$ s.t. $w_i \geq \frac{W}{2}$. We consider the 2nd ($B=1$) mode

Then, with probability $\frac{1}{2}$, this item will be packed.

$$\Rightarrow \mathbb{E}[ALG] \geq \frac{1}{2} \cdot \frac{1}{2}W = \frac{1}{4}W.$$

$OPT \leq W$, so,

$$\Rightarrow \frac{OPT}{\mathbb{E}[ALG]} \leq 4. \quad \#.$$