Online Algorithms

$$\text{Competitive Ratio}(A) = \max_{\sigma} \frac{Alg(\sigma)}{OPT(\sigma)} \leftarrow \text{strict competitive ratio}$$

Algo is (weakly) c-competitive if $Alg(\sigma) \leq c.OPT(\sigma) + O(1)$ for $\forall \sigma \in$ inputs.

Unless otherwise specified, we'll look at strict CRs in this lecture

---

Oblivious (for Randomized Algos) = fix the sequence up front, or adversary does not see the algos coin tosses

$$CR = \frac{E(Alg(\sigma))}{OPT(\sigma)}$$

Rent-or-Buy: (Ski-Rental) (turn off processor into low power, etc)

Rent for \$1 per day

Buy for \$B forever.

Sequence : $\underbrace{YYY \ldots Y}N$ ___ from then on.

Optimal strat: if @ day B still Y then buy on that day, rent until then.

$$C.R(\sigma) = \frac{\text{if length} < B \text{ then len else } 2B-1}{\min(\text{len}, B)} \Rightarrow \max_{\sigma} = \frac{2B-1}{B} = 2 - 1/B.$$

And since algos are simple (rent until day i and then buy)
can check that any other algo is strictly worse.

Random:

Facts 1: Sps length $\geq B$ then can assume length $= \infty$
OPT$(B+x)$ = OPT(∞) so we can only hurt ourselves more.
$\Rightarrow$ we should buy on some day in $[0..B]$
and length will only come from $[0..B] \cup \{\infty\}$
and 0 is useless

Player's strategy : buy on day 1. . . . buy on day B

Adversary : length $= 1$, len $= 2$ . . . len $= \{B = \infty\}$

So let's consider B = 4. length

begin day

	1	2	3	4
1	4/1	4/2	4/3	4/4
2	1/1	5/2	5/3	5/4
3	1/1	2/2	6/3	6/4
4	1/1	2/2	3/3	7/4

} each other row has some value > 7/4.

this is ~~cost~~ optimal deterministic

Randomized Algo is distribution over deterministic strategies.

p_1, p_2, p_3, p_4 (say) Say set these so that payoff against each col = C

{ should ~~max~~ have ≤ C and minimize C but we luck out here. }

$$\Rightarrow \quad 4p_1 + p_2 + p_3 + p_4 = 1.C$$
$$4p_1 + 5p_2 + 2p_3 + 2p_4 = 2C$$
$$4p_1 + 5p_2 + 6p_3 + 3p_4 = 3C$$
$$4p_1 + 5p_2 + 6p_3 + 7p_4 = 4C \qquad p_1 + p_2 + p_3 + p_4 = 1.$$

$$\Rightarrow \quad 4p_1 + p_2 + p_3 + p_4 = C \qquad \Rightarrow p_4 = C/4$$
$$4p_2 + p_3 + p_4 = C \qquad p_3 = (3/4)\, C/4$$
$$4p_3 + p_4 = C \qquad p_2 = (3/4)^2 C/4$$
$$4p_4 = C \qquad p_1 = (3/4)^3 C/4$$

sum to 1

$$\Rightarrow C = \frac{1}{1-(1-\frac{1}{4})^4}$$

in general $C = \dfrac{1}{1-(1-\frac{1}{B})^B} \approx \dfrac{e}{e-1}$.

Can write an LP as well, but this is going to give the same answer.

Also: to check if this is indeed opt (since we just set all equal), solve for column player. Will get the same answer (check!).

Paging

k pages in Cache:

Fact: No online algorithm (deterministic) can be $<k$ competitive.

Pf: $k+1$ pages in all. Each time ~~evict~~ request page not in cache. Alg $= n$.
But OPT strategy can ~~request the~~ evict farthest item in the future,
one eviction per k requests. $\Rightarrow$ gap $\geq k$. ∎

Fact: LRU, FIFO are both k-competitive. In fact 1-bit LRU is also.

Pf: Maintain 1 bit per page. If used, mark. If need to evict ~~evict~~ evict unmarked page. ~~Epoch~~ if all pages in cache marked, and new page requested not in cache, unmark all of them. (End of ~~epoch~~ phase, start new ~~epoch~~ phase)
↓ (and mark page brought in)

In an ~~epoch~~ phase, we pay $\leq k$. Opt pays ≥ 1 (since $k+1$ distinct requests). ~~BTW, if~~

$\Rightarrow$ all these are optimal (Proofs for LRU, FIFO omitted).

Randomized: Random Marking is $O(\log k)$ competitive

Random Marking: ① 1 bit per page. (initially all unmarked)
② If request not in cache, evict random unmarked page.
bring new page in, mark it
③ if all pages marked and request come in for another page,
Unmark all and go to ②.
↑ start new phase.

Let S_i = set of pages in our cache at start of ~~epoch~~ phase i

$\Delta_i := |S_{i+1} - S_i|$.

Fact: E[# of evictions in phase i] $\leq \Delta_i (H_k + 1)$

Pf: look at distinct requests in phase i ← all stay in cache until end of phase.

clean: request for page not in S_i — stale: page was in S_i

↑ we pay 1, but OPT also pays 1. — ↑ opt may pay 0, we may pay 1.

Sps c clean requests so far, s stale, $s+1^{st}$ stale comes in.

$\Pr[\text{it causes a cache miss}] = \frac{c}{k-s}$ ← we've evicted c random pages ← out of so many pages

$\Rightarrow$ E[cache misses due to stales] $\leq \frac{c}{k} + \frac{c}{k-1} + \cdots \leq c_{final} H_k \leq \Delta_i \cdot H_k$

misses due to cleans $\le c_i \,\cancel{\le}\, \underset{\uparrow}{=} \Delta_i$. $\Rightarrow E[\#\text{misses}] \le \Delta_i(H_k + 1)$.

Fact: $OPT \geq \sum_i \Delta_i$.

Pf: Let Φ_i = #pages in S_i but not in OPT (random variable). at that time.

- OPT must pay at least $c_i - \Phi_i$ in phase i
  it can have Φ_i extra pages that are in cache.
  b/c $|S_i \Delta OPT_i| = \Phi_i$

- ~~OPT~~ at end of phase i, Alg has k most recent requests,
  OPT doesn't have Φ_{i+1} of them $\Rightarrow$ cost(OPT) in phase i $\geq \Phi_{i+1}$

$$\Rightarrow C_{OPT}(i) \geq \max\{\Phi_{i+1}, c_i - \Phi_i\} \geq \frac{c_i - \Phi_i + \Phi_{i+1}}{2}$$

$\Rightarrow$ total cost of OPT $\geq \frac{1}{2}(\sum c_i - \Phi_{init} + \Phi_{final})$

$\geq \frac{1}{2}\sum c_i$

$\Phi_{init} = 0$

$\Phi_{final} \geq 0$

☑

$\Rightarrow E[Alg] \leq O(\sum c_i H_k)$
$= o(OPT \log k)$.

Thm: Best possible. ~~take k~~ Use Yao principle — lower bound for randomized algo comes from showing a distribution on which any deterministic algo does poorly.

One such distrib: $k+1$ pages, k cache size, each request for random (uniform) page. Alg makes miss $\simeq \frac{1}{(k+1)}$ fraction of time. $\Rightarrow \simeq L/(k+1)$ cost.

OPT: length of ~~epoch~~ phase (when you see $k+1^{st}$ distinct request)
$\simeq$ coupon collector $\Rightarrow (k+1)H_{k+1} \Rightarrow$ # of phases $\simeq L/(k+1)H_{k+1}$

opt pays at least once per phase

$\Rightarrow$ gap of H_{k+1}.

Online set cover: (fractional).

Sets $S_1, S_2 \dots S_m \subseteq U$. $|U| = n$.

Elements from U arrive online. Pick sets fractionally to cover elements.

But must be monotone.

i.e. Solve $\min \sum x_S$

elements until time t.

$$\text{st } \sum_{S \ni e} x_S \geq 1 \quad \forall \text{ elements that arrive } e \in E_t$$

$$x_S \geq 0.$$

and x_S increase over time. [N.b. constraints of this LP arrive over time].

dual: $\max \sum_{e \in E_t} y_e$

$$\text{st } \sum_{e \in S} y_e \leq 1 \quad \forall S \in \mathcal{F}$$

$$y_e \geq 0.$$

Primal dual: ~~Start with~~ $x_S \leftarrow 0$ at beginning.

```
when element $e_t$ arrives :-
    while $\sum_{S \ni e_t} x_S < 1$ then
        {$\forall S: e_t \in S$, do $x_S \leftarrow 2x_S + \frac{1}{m}$
```

$y_{e_t} \leftarrow y_{e_t} + 1$ for each update on the primal

---

Analysis: ① each update, $\Delta P \leq 3 \Delta D$.

② P is feasible at end

③ D is not feasible, but $D/(\log m)$ is feasible.

$\Rightarrow$ final primal value $= \sum_t \Delta_t P \leq 3 \sum_t \Delta_t \text{Dual} = 3$ final dual

but since $D/(\log m)$ is feasible $\Rightarrow D \leq O(\log m)$. OPT primal/dual.

$\Rightarrow$ final primal $\leq O(\log m)$ OPT primal.

Proof of facts: when update,

① $\underline{\Delta P}$: $\Delta \sum_{S\text{ charged}}' x_S = \sum_{S: e_t \in S}' \Delta x_S$

$\Delta x_S = (2x_S + \frac{1}{m}) - x_S$

$= x_S + \frac{1}{m}$

$= \sum_{S: e_t \in S}' x_S + \sum_{S: e_t \in S}' \frac{1}{m} < 1 + 1. = 2$

$= 2\Delta D.$

↑ was unfeasible ↑ m sets.

② P feasible by construction

③ Everytime dual variable for element in set S increased, $x_S \uparrow 2x_S + 1/m$.

$\Rightarrow$ first time $x_S \leftarrow 1/m$.

then it doubles. Can only happen $\log_2 m$ times before this set ≥ 1 and can never be in a violated constraint

---

Rent or Buy using P-D.

$$\min \sum_{t=1}^{\infty}{}' x_t + yB \qquad \max \sum_t z_t$$

st $\quad x_i + y \geq 1 \quad \forall i = 1, 2 \dots$

$x_i, y \geq 0.$

st $\quad z_t \leq 1 \quad \forall t$

$\sum' z_t \leq B.$

$z_t \geq 0.$

Algo: each day $\begin{cases} y \leftarrow \alpha y + \beta \\ x_t \leftarrow 1 - y \end{cases}$ and $z_t \leftarrow 1.$

unless $y \geq 1$ already.

- $\Delta P = \cancel{(\alpha-1)yB + \beta}$ ~~$(\alpha+\beta-1)$~~ $\quad \Delta D = 1.$
  $B((\alpha-1)y + \beta) + 1$ ~~$(\alpha-1)B+\beta$~~
- Primal is feasible ✓
- Dual: we'll stop at t days
  where $t = \ln_\alpha \left\{\frac{\alpha+\beta-1}{\beta}\right\}$

$y_1 = \beta.$

$y_2 = \alpha\beta + \beta.$

$y_3 = \alpha^2\beta + \alpha\beta + \beta$

$\Rightarrow y_t = \frac{(\alpha^t - 1)}{(\alpha - 1)} \cdot \beta. \geq 1$ (say)

$t = \ln_\alpha \left\{\left(\frac{\alpha-1}{\beta}\right) + 1\right\}$

Say: $\alpha = (1 + \frac{1}{B})$

$\beta = \frac{1}{CB} \Rightarrow \ln_\alpha \alpha = \frac{1}{\log \alpha}$

Rent or Buy using P-D

$$\min \sum_1 x_t + yB$$

$$x_t + y \geq 1 \quad \forall t$$

$$x_t, y \geqslant 0.$$

$$\max \sum z_t$$

$$\text{st} \quad z_t \leq 1$$

$$\sum_1 z_t \leq B.$$

$$z_t \geqslant 0.$$

Update:
$$\begin{cases} y \leftarrow (1+\frac{1}{B})y + \frac{1}{CB} \\ x_t \leftarrow 1-y \\ z_t \leftarrow 1 \end{cases}$$

① Δ Primal $= 1-y + B\left[(1+\frac{1}{B})y + \frac{1}{CB} - y\right] = 1-y+y+\frac{1}{C} = 1+\frac{1}{C}$

Δ Dual $=1$

$\Rightarrow$ ratio : $(1+1/C)$

② Primal feasible ✓

③ Dual feasible (need to set C):-

after t days, y becomes $\alpha^{t-1}\beta + \alpha^{t-2}\beta + \dots \alpha\beta + \beta$

where $\alpha = 1+\frac{1}{B}$

$\beta = \frac{1}{CB}$

$$= \frac{(\alpha^t - 1)}{(\alpha - 1)} \cdot \beta$$

$$= \frac{(1+\frac{1}{B})^t - 1}{1/B} \cdot \frac{1}{CB} = 1$$

$\uparrow$

~~set~~ want this for t=B for dual feasible

$$\Rightarrow (1+\frac{1}{B})^B - 1 = C$$

$$\Rightarrow 1+\frac{1}{C} = 1 + \frac{1}{(1+\frac{1}{B})^B - 1}$$

$$= \frac{(1+\frac{1}{B})^B}{(1+\frac{1}{B})^B - 1} \simeq \frac{e}{e-1}$$