Lecture 11

L11

Randomization in Optimization under Uncertainty

today

1. Picking the max value on line
   - the Secretary (random order model)
   - the Prophet (stochastic model).
2. Extensions to picking k items
3. Extensions to other structures.

— x —

Another thread: Online learning (if time permits)

maybe another day

A. Choosing from Expert Advice
   - regret minimization

B. Multi-Armed Bandits
   - Adversarial settings
   - stochastic settings (i.e. compete with best coin)

today.

— x —

Prophets & Secretaries

Making decisions in the face of uncertainty — with the help of randomization!

Suppose we see a sequence of n numbers (non-negative) reals.

$a_1, a_2 \ldots a_n$.

But we have no idea of the numbers $a_{i+1} \ldots a_n$ when we see a_i

(future is uncertain).

Want to pick the max number. (with high prob, if randomized)

(can only pick one number. Game ends when we pick anyone)

"when we hire one secy".

———— x ————

Worst case:

- deterministic algos. can do nothing. (even when $n=2$)

  - sps first number $a_1 = 1$.

    Algo picks a_1 — then give $a_2 = M >> 1$.

    or

    Algo does not pick a_1 — then give $a_2 = 1/M << 1$.

- Even randomization by itself does not help. all that much.

  - $\exists$ Algo which is n-approximate (i.e. $P(\text{pick max}) \geq 1/n$).

    - Algo picks a random index I uniformly from $\{1..n\}$.

      picks element a_I. $\Pr[a_I \text{ max}] \geq 1/n$.

  - Can we do better?

    - Sadly no. (Yao's Lemma)

So instead, consider the following models:— Extra Assumptions! (on the input)

(1) We are given n random variables.

$X_1, X_2 \dots X_n$. ← these are indep r.vs.

(i.e. we know their distributions)

And $a_i \sim X_i$. (independently) ← Useful in auctions!

Now can we do better?

This is called the "Prophet" model.

Ideal theorems:

$\mathbb{E}[\text{value picked}] \geq \alpha \cdot \mathbb{E}[\max(X_1, X_2 \dots X_n)]$

↑ "prophet's value"

↑ expectation over both randomness in $X_1, X_2 \dots X_n$ and the algo. (if any).

———— x ————

(2) Adversary picks n numbers. $b_1, b_2 \dots b_n$.

Then "nature" picks a random permutation $\sigma: [n] \to [n]$.

And the number $a_i = b_{\sigma(i)}$

so we see the numbers in random order (unif from all $n!$ perms).

Ideal theorems:

$\Pr[\text{pick max}] \geq \alpha$. "ordinal"

or

$\mathbb{E}[\text{value picked}] \geq \alpha \cdot \max(b_1, b_2 \dots b_n)$. "cardinal".

Called the secretary model.

Let's start easy:

Get a constant for the secretary problem:— (Assume distinct values.)

Algo 1: "Learn-and-Pick".

Observe the first $n/2$ items, do not pick.

Let $\tau = \max(a_1, a_2 \ldots a_{n/2})$.

Pick the first item in $n/2+1 \ldots n$ with value $> \tau$.

Claim: $\Pr(\text{pick max}) \geq 1/4$.

Pf: Consider events

$\mathcal{E}_1$ = second highest item in first half of elts.

$\mathcal{E}_2$ = highest item in second half.

$\{\text{pick max}\} \supseteq (\mathcal{E}_1 \cap \mathcal{E}_2)$

$\Rightarrow \Pr(\text{Pick max}) \geq \Pr(\mathcal{E}_1)\,\Pr(\mathcal{E}_2 \mid \mathcal{E}_1)$ → $\Pr(\mathcal{E}_2)$.

$\geq 1/2 \cdot 1/2 = 1/4$.

—— x ——

Actual optimal strategy:—

Wait for n/e timesteps $\cong 0.37n$.

t = max in 1st $1/e$ fraction

Pick 1st item after that which is larger.

Dynkin Algo or 37% rule

$\Pr(\text{Dynkin's algo picks max}) \geq 1/e \cong 0.37$. (Proof a bit more involved.)

Let's do prophets now

recall: we know the distributions of $X_1, X_2 \ldots X_n$, (independent rvs) and $a_i \sim X_i$

Algo (one of many)

Median Algorithm [Kleinberg Suchetma Gatling]

def $X_{max} = \max(X_1, X_2 \ldots X_n)$

def τ = median (X_{max})

i.e. $Pr(X_{max} > \tau) = Pr(X_{max} < \tau) = 1/2$.

Assume for simplicity: X_i's are continuous and there are no point masses

Only for simplicity

Can compute τ up-front (information theoretically, at least) because we know the distributions

Algo: pick first item $\geq \tau$.

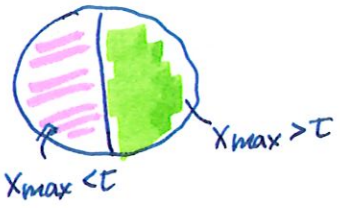
Thm: $\mathbb{E}[Alg] \geq \frac{1}{2}\mathbb{E}[X_{max}]$

Intuition for $1/4$

- note: $Pr(X_{max} < \tau) = 1/2 \Rightarrow$ wp $1/2$ no item is chosen

  $\Rightarrow$ wp $\geq 1/2$ we have not picked anyone at any position i

-



$$E[X_{max} \wedge (X_{max} > \tau)] \geq E[X_{max} \wedge (X_{max} < \tau)]$$

$$\geq \frac{1}{2} E[X_{max}]$$

$\Rightarrow$ can use these intuitions to get $E[Alg] \geq \frac{1}{4} E[X_{max}]$.

Here's the tighter (and actually easier) analysis.

$(a)^+ := \max(a, 0)$

- $\mathbb{E}[Alg] \geq \tau \cdot \Pr(X_{max} > \tau) + \sum_{i=1}^{n} \mathbb{E}[(X_i - \tau)^+] \Pr(X_j < \tau \ \forall j < i)$

$$\geq \tau \cdot \underbrace{\Pr(X_{max} > \tau)}_{\geq 1/2} + \sum_{i=1}^{n} \mathbb{E}[(X_i - \tau)^+] \cdot \underbrace{\Pr(X_{max} < \tau)}_{\geq 1/2}.$$

$$= \frac{1}{2}\left[\tau \cdot + \sum_{i=1}^{n} \mathbb{E}[(X_i - \tau)^+] \cdot\right]$$

- $\mathbb{E}[X_{max}] \leq \tau + \mathbb{E}((X_{max} - \tau)^+)$

$$\leq \tau + \sum_{i=1}^{n} \mathbb{E}((X_i - \tau)^+)$$

$\Rightarrow \mathbb{E}[Alg] \geq \frac{1}{2} \cdot \mathbb{E}[OPT]$. ▨

This is the best possible.

Sps. $X_1 = 1$ wp 1.

$$X_2 = \begin{cases} 0 & \text{wp } 1-\varepsilon \\ 1/\varepsilon & \text{wp } \varepsilon. \end{cases} \quad \Rightarrow \mathbb{E}X_2 = 1 \ .$$

What can alg do after seeing $X_1 = 1$. ?

Pick X_1? $\Rightarrow$ get value 1.

Reject it, pick X_2 $\Rightarrow$ get $\mathbb{E}[\text{value}] = 1$

but $X_{max} = \begin{cases} 1 & \text{wp } 1-\varepsilon \\ 1/\varepsilon & \text{wp } \varepsilon \end{cases}$

$\Rightarrow \mathbb{E}[X_{max}] = 2 - \varepsilon.$

What if we want to pick k items, or change other parts of the setup?

E.g. **k-item prophet**. Again $X_1, X_2 \dots X_n$ independent r.vs.

Pick at most k items S s.t.

$$\mathbb{E}\left[\sum_{i \in S} X_i\right] \geq \alpha \cdot \mathbb{E}\left[\max_{\substack{T \subseteq [n] \\ |T| \leq k}} \sum_{i \in T} X_i\right].$$

Above proof seems tricky to generalize.

Let's give a different proof based on Linear Programs,
(the "ex-ante relaxation").

For this LP method, assume (for simplicity) that $X_i \in \{v_1, v_2 \dots v_M\}$ $\forall i$
for some finite set of values (at most M values in support of X_i)

Start with just single-item prophet.

Fix the optimal algorithm (as a thought experiment).

let $y_i(v)$ = Probability that we select item i and it takes on value v when we select it.

Some constraints:

- $\sum_{i,v} y_i(v) \leq 1$ — at most one item picked
- $y_i(v) \leq P_i(v)$ — we can pick item i at value v only if i takes on value v.

Alg's value: $\sum_{i,v} v \cdot y_i(v)$.

$y_i(v) \geq 0$

Of course, there could be other constraints on the optimal Alg's value, but we can solve this "ex-ante" LP and hence get a value s.t.

Fact 1: $OPT \le LP$.

Now: we will give an algo such that

$$E[Alg] \ge \frac{1}{4} LP \ge \frac{1}{4} OPT.$$

LP (recall):

$$\max \sum_{i,v} v \cdot y_i(v) \quad \text{(vars of LP)}$$

$$\text{s.t } y_i(v) \le P_i(v)$$

$$\sum_{iv} y_i(v) \le 1$$

also $y_i(v) \ge 0$

Algo:

- Solve LP. Up front, based only on distributions of $X_1, \ldots X_n$
- When we see item i.
  - Suppose it takes on value v
  - pick it with prob $\frac{y_i(v)}{2P_i(v)}$. (if no item picked earlier).

Lemma 1: $\Pr(\text{no item picked in entire run}) \ge 1/2$.

Proof: $E[\#\text{items picked}] \le \sum_{iv} P_i(v) \cdot \frac{y_i(v)}{2P_i(v)} = \frac{1}{2}\sum y_i(v) = 1/2$.

$\Rightarrow \Pr(\text{at least one item picked}) \le E[\#\text{items picked}] \le 1/2$

↑ first moment method.

$\Rightarrow \Pr(\text{no item picked by position } i) \ge 1/2$.

Lemma 2: $E[\text{value of Alg}] \ge \frac{1}{4} LP$

Pf: $E[Alg] = \sum_{iv} v \cdot \Pr(i \text{ has value } v) \cdot \Pr(\text{we still have room}) \cdot \Pr(\text{pick item})$

$$\ge \sum_{iv} v \cdot P_i(v) \cdot 1/2 \cdot \frac{y_i(v)}{2P_i(v)} = \frac{1}{4}\sum v \cdot y_i(v) = \frac{1}{4} LP$$

☺

So putting it all together:

$$E[Algo] \geq \frac{1}{4} E[OPT]$$

□

We can get a better rounding algo from the LP that shows

$$E[Algo] \geq \tfrac{1}{2} LP \geq \tfrac{1}{2} OPT.$$

[See work by Alaei]

And LP approach is quite general!

E.g. suppose we want to select K items.

new LP: $\max \sum_{i,v} v \cdot y_i(v)$

s.t. $y_i(v) \leq P_i(v)$

$$\sum_{i,v} y_i(v) \leq K$$

$y_i(v) \geq 0$.

Fact: $OPT \leq LP$

Again: Algo = if item i takes on value v then pick up. $\frac{y_i(v)}{P_i(v)} \cdot (1-\varepsilon)$. (if not at capacity already)

Lem: $E[\#\text{items picked}] = \sum_{i,v} Pr(i \text{ takes value } v) \cdot Pr(\text{we pick it})$

$$= \sum P_i(v) \cdot \frac{y_i(v)}{P_i(v)} (1-\varepsilon) \leq K(1-\varepsilon).$$

And by a chernoff bound, $Pr(\#\text{items picked} \geq K) \leq \exp\left(-\frac{\varepsilon^2 K}{\text{const}}\right)$.

Which, if $\varepsilon = c\sqrt{\frac{\log k}{k}}$ is at most $\frac{1}{\text{poly}(k)}$.

$\Rightarrow$ whp we pick $< k$ items at end

$\Rightarrow$ whp we pick $\leq k$ items at any time

$\Rightarrow$ whp we don't run out of capacity at step i

$$\Rightarrow E[\text{value}] = \sum_i \sum_v v \cdot Pr(i \text{ takes value } v) \cdot Pr(\text{we pick } i)$$

$$= \sum_i \sum_v v \cdot p_i(v) \cdot \frac{y_i(v)}{p_i(v)} \cdot (1-\varepsilon) \cdot \left(1 - \frac{1}{\text{poly}(k)}\right)$$

$$\geq LP\left(1 - \varepsilon - \frac{1}{\text{poly}(k)}\right) = LP\left(1 - c\sqrt{\frac{\log k}{k}}\right)$$

☒

Very versatile idea of using the LP.

Can extend to other settings

Have n r.vs.. can inspect values of at most B of them

can then pick at most K of them.

Max value.

Similar ideas

- solve LP
- use that to define policy
- show that policy gets large value.

More general settings:

- Want to pick edges of a graph so that they are acyclic and maximizes total weight (apx).

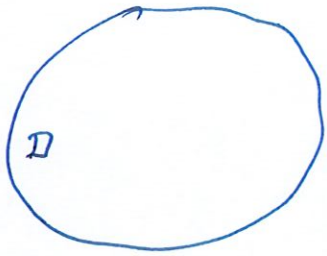
  the graphical "matroid"

  get O(1) apx.

- Given a graph with k sources.

  Clients are other vertices.

  Pick a max weight set of clients that can be connected to s without violating edge capacities



(called a "gammoid").

The "Netflix" "matroid"

Q: can we get O(1) apx?

can do O(log log n) apx.

Lots of research in Prophets & Secretaries

(Auctions are a major application).

See online for details.

But for now: Lets briefly talk about bandits.

(Stochastic) Multi-Armed Bandits

There are k slot machines / "bandits" / arms.

Each of them, when pulled, give \$1 wp p_i (for i^{th} arm) independently.

(So, playing arm i at any time $\Leftrightarrow$ flipping coin wp p_i)

We don't know these probabilities.

Want a strategy that plays arms over time so that over T timesteps

$$-\mathbb{E}[Alg] + T \cdot p_{max} =: \text{regret}$$

is small.

def: $\Delta_i = p_{max} - p_i$

Thm: $\exists$ strategy with regret $\leq \sum_{i=2}^{k} \frac{O(\log T)}{\Delta_i}$

Many algorithms give this result —

let's see two of them

1. Active Arm Elimination (AAE)
2. Upper Confidence Bound (UCB)

Active arm Elimination (Assume T is known for now)

$S \leftarrow \{1, 2 \ldots k\}$ initially

↑ Active Arms

for $i = 1, 2 \ldots$

set $\varepsilon_i = 2^{-i}$

for each arm in S

play it $O(\frac{1}{\varepsilon_i^2} \log T)$ times

Let $\hat{p}_j^i$ be the sample mean of arm $j \in S$ (in this round i samples)

Remove from S all arms j s.t. $\hat{p}_j^i < (\max_{j' \in S} \hat{p}_{j'}^i) - 2\varepsilon_i$

(Eliminate)

—— x ——

Fact: whp $(1 - \frac{1}{\text{poly}(T)})$ have that $\hat{p}_j^i \in p_j \pm \varepsilon_i$ ↙ chernoff bound.

↑ let's condition on all these good events happening

⇒ ① best arm not eliminated (say $p_1 > p_2 \geq p_3 \cdots \geq p_k$)

Indeed: $p_1 > p_j$

$\Rightarrow \hat{p}_1 + \varepsilon_i > \hat{p}_j - \varepsilon_i \Rightarrow \hat{p}_1 > \hat{p}_j - 2\varepsilon_i$ $\forall j$

⇒ elimination condition never sat'd for p_1.

② All arms with $p_j < p_1 - 4\varepsilon_i$ eliminated

Indeed: $\hat{p}_j - \varepsilon_i < (\hat{p}_1 + \varepsilon_i) - 4\varepsilon_i \Rightarrow \hat{p}_j < \hat{p}_1 - 2\varepsilon_i$

⇒ Lemma: Arm j only survives until round i

s.t. $p_1 - p_j = \Delta_j \geq 4\varepsilon_i$

Now arm j is played in rounds $i = 1, 2, \ldots, i(j)$ s.t. $\varepsilon_{i(j)} \le \Delta_j/4$

$$\Rightarrow \quad N_j = \#\text{times played} = \Theta\left(\frac{\log T}{\varepsilon_1^2} + \frac{\log T}{\varepsilon_2^2} + \cdots + \frac{\log T}{\varepsilon_{i(j)}^2}\right)$$

$\nearrow$

conditioned on good event $\quad = \Theta\left(\frac{\log T}{\Delta_j^2}\right)$

To remove conditioning: $E[N_j] = \left(1 - \frac{1}{\text{poly}(T)}\right) \cdot O\left(\frac{\log T}{\Delta_j^2}\right) + \frac{1}{\text{poly}(T)} \cdot T$

$$\le O\left(\frac{\log T}{\Delta_j^2}\right) + 1.$$

$\Rightarrow$ E[regret] due to playing the suboptimal arms

$$= \sum_{j \ge 2}^{} \Delta_j \left(O\left(\frac{\log T}{\Delta_j^2}\right) + 1 \right) = \sum_{j \ge 2}^{} \left(\Delta_j + \frac{O(\log T)}{\Delta_j} \right)$$

If Δ_j's are small, we want to get a different bound:—

$$\text{Regret} \le O\left(\sqrt{kT \log T}\right).$$

Pf: $\sum_j \Delta_j E[N_j] = \sum_j \sqrt{\Delta_j^2 E[N_j]} \sqrt{E[N_j]}$

$$\le \sqrt{\sum_j \Delta_j^2 E[N_j]} \sqrt{\sum_j E[N_j]}$$

Cauchy-Schwarz

$$\le \sqrt{K \log T} \cdot \sqrt{T}$$

UCB: Upper Confidence Bound Algo.

Again, same setting. Assume $p_1 \geq p_2 \geq \cdots \geq p_K$ and $\Delta_j = p_1 - p_j$

Again, assume we know length of experiment T.

Algo:

- Set $N_j = 1 \ \forall j$. pull each arm once.
- at time step t,

define $\hat{p}_j$ = $\frac{\#\text{times we've seen heads when pulling } j}{N_j}$

define $\tilde{p}_j = \hat{p}_j + \sqrt{\frac{c \cdot \log TK}{N_j}}$ (upper confidence bounds)

pick j with largest $\tilde{p}_j$, pull it, increment N_j

Fact 1: $\Pr\left(\hat{p}_j \in p_j \pm \sqrt{\frac{c \log TK}{N_j}} \quad \forall \text{ times } t \ \forall \text{arms } j\right) \geq 1 - \frac{1}{T^2}$.

Fact 2: if above "good event" $\Rightarrow \tilde{p}_j \geq p_j$.

Fact 3: if j is argmax at time t, then $\tilde{p}_j \geq \tilde{p}_1 \geq p_1$

$$\Rightarrow p_j + 2\sqrt{\frac{c \log TK}{N_j}} \geq p_1 = p_j + \Delta_j$$

$$\Rightarrow N_j \leq \frac{O(\log TK)}{\Delta_j^2} \quad \text{in this case.}$$

of course, w.p. $1/t^2$ could have N_j finally be $\leq T$

$$\Rightarrow \quad E[N_j] \leq 1 + \frac{O(\log(KT))}{\Delta_j^2}$$

$$\Rightarrow \text{again regret} = \sum_{j=2}^{K} \Delta_j \, E[N_j] \leq \sum_{j \geq 2} \left(\Delta_j + \frac{O(\log KT)}{\Delta_j} \right)$$

And same Cauchy Schwarz gives $O(\sqrt{KT \log KT})$

(Can probably remove the K from inside the log term, but small improvement)

UCB is idea of "optimism in the face of uncertainty".

Basically: if we pull arm few times, wide confidence interval

$\Rightarrow$ incentivized to pull arms with

(a) large $\hat{p}_j$

(b) large confidence intervals $\Leftrightarrow$ few pulls so far.

---

lots of places in optimization under uncertainty

Where randomness helps alot!

"Experts": predictions online

"Online Algorithms": decision making online

"Distributed Algorithms": each machine makes local decisions etc.

See textbook or ask for more details!