today: limitations of learning

NP hard to achieve small regret

or ~~doing~~ doing one step of

follow the leader is NP-hard even

approximately

Model: i values v_{ij} for item j

$$v_i(A) = \max_{j \in A} v_{ij}$$

We know: if not regretting (all player)

bidding $(0, \dots 0, \frac{v_{ij}}{2}, 0 \dots 0)$ $\forall j$

$\Longrightarrow$ Social welfare $\geq \frac{1}{2}$ OPT

bidding $(0 \dots 0, k\delta, 0 \dots 0)$ of this form

$\#$ strategies $= \#$ items $\cdot \frac{1}{\delta} = K$

Example today:

Your value $v_{ij} = v$ all j

Opponent bid: $\underline{1} \ll v \ll \underline{v'}$

on all items

Made with Goodnotes

Sets to opponent bids 1 are

T_1, or T_2 or ... or T_m equally likely

non-empty

bidding rationally:

bid 1^+ on a set S & 0 all others

resulting value (expected)

$$u(S) = \frac{1}{m} \sum_i \left(v \,|T_i \cap S \neq \emptyset\right) - |T_i \cap S|$$

price

Suppose $\max_i |T_i| = d$

$v = 2dm$

<u>Claim</u> optimal (near optimal) bid

ensures winning an item

<u>Proof</u>: e.g. $S = \bigcup T_i$

this gets you: $v - \frac{1}{m} \cdot \sum_i |T_i|$

if S does not guarantee a win

$$\Longrightarrow u(S) \leq v \cdot \frac{m-1}{m} \doteq v - 2d$$

Recall Hitting Set (NP-hard)

$T_1, \dots T_m$ non-empty sets

find S s.t. $S \cap T_i \neq \emptyset$ all i

& $|S|$ minimal

<u>Our problem</u>

$T_1 \dots T_m$ non-empty

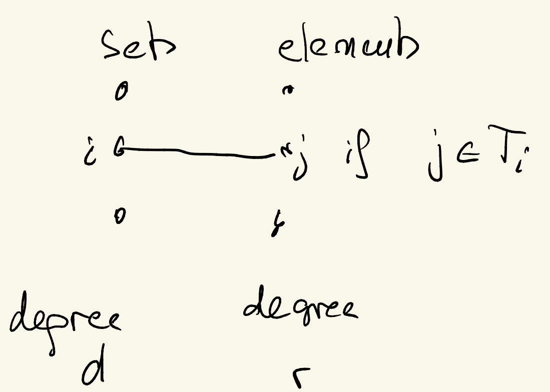
find S s.t. $T_i \cap S \neq 0$ all i

& $\sum_i |T_i \cap S|$ minimal.

---

variant: regular hitting set also NP-hard

$|T_i| = d$ all i & $|\{i : j \in T_i\}| = r_j = r$ all j



NP-hard to even approximate to a factor better than $\frac{1}{2} \ln r$

goal find hitting set S

min $|S|$

Our objective is induct $T_1 \dots T_m$ is part of regules hitting set.

$$u(S) = v - \frac{1}{m}\sum_i |S \cap T_i| = v - \frac{1}{m}\sum_{j \in S} r_j = \frac{r|S|}{m}$$

hitting set

All poly time algorithm must have regret due to picking $|S'| \geq \frac{1}{2} \ln r \cdot |S|$

Recall regret in our algorithms

$$O\left(\sqrt{\ln K \cdot T}\right) = O\left(\sqrt{n \log \tfrac{1}{\delta} \cdot T}\right)$$

$K = \#\text{strategies} = \left(\frac{1}{\delta}\right)^n$

$(k_1\delta, \dots, k_n\delta)$ for n items

Multiplicative weight: too many probabilities to maintain to run

Follow the leader: cannot implement on step.