15854(B): (Advanced) Approx Algos

- HWS. · 10 weeks. Xinyu / Praveen / Anupam.
- Piazza / Gscope / etc.

- Optimization Problems. ~~[illegible]~~ vs. Decision Problems.

  - ~~Max~~ 3SAT. vs. Max 3SAT, min 3-unsat etc.
  - 2SAT (in P) vs. Max 2SAT, min 2-unsat -
  - 3color (NP-comp) vs. Min-Coloring, Max 3 cut etc.
    2 color (in P) vs. Max 2cut, Min 2 uncut.

- ~~Landscape is~~ Approx Problems
  - Input: instance I.
  - ~~output~~ · valid solution $S. \in sols(I)$.

    assume: checking S is in P. Also that there exists a soln

    notion of value for each sol^n S. $val_I : sols(I) \to \mathbb{R}_{\geq 0}$.

    - minimization or max.

  Want: algo: maps instances to solutions st. $val(S(I))$ is as high or low as possible.

Example: Max 3SAT. Instances are 3CNF formulas. (m clauses, n vars)

solutions: any truth assignment, assignment of T/F to each var.

value : $\dfrac{\#\text{ satisfied clauses}}{\text{total }\#\text{clauses}} \in [0,1]$.

Algo: assign random T/F assignment.

Produces solution that satisfies 7/8 clauses in expectation

Can we do better?

$\boxed{\text{Approx ratio} = \text{worst case value } \frac{Alg(I)}{Opt(I)}}$

Finer grained Landscape than just NP-hardness

1. Min-TSP ($\mathbb{R}^2$) is NP hard. but has PTAS. — later
2. Min-Set cover is NP-hard but has $O(\lg n)$ apx. — soon
3. Max 3SAT ——— but $7/8$ apx is best possible
4. Max Clique ——— but $n^{1-\varepsilon}$-apx for any $\varepsilon > 0$ $\Rightarrow P = NP$.

So goal of course: apx algos (show positive results)

hardness of apx (negative results)

or integrality gaps / algorithmic gaps (limitations of our algos).

Sometimes talk about c-vs-s search problems. $s \le c$

[if $OPT(I) \geq c$ produce soln of value $\geq s$]

or c-vs-s decision problems.

[if $OPT(I) \geq c$, answer YES, else if $OPT(I) < s$, answer NO]
(anything in the middle is OK)

<u>Fact</u>: Algo for c-vs-s SEARCH $\Rightarrow$ c-vs-s DECISION

<u>Pf</u>: ~~run algo, if~~ ~~[illegible]~~, ~~[illegible]~~ gah!

~~OPT(I) ≥ c then ALG(I) ≥ s (so output YES if ALG(I) ≥ s)~~

~~if ALG(I) < s ⇒ OPT(I) < c too so answer NO is OK.~~

Run algo. if $ALG(I) \geq s$ say YES, else say NO.

$\Downarrow$ $OPT(I) \geq s \Rightarrow$ YES is fine

$\Downarrow$ $OPT(I) < c \Rightarrow$ NO is fine.

min Set Cover : General problem, captures many others. (HW?)

Input : Set system $\mathcal{F} = (\mathcal{U}, \{S_1, S_2 \dots S_m\})$ $S_i \subseteq \mathcal{U}$.

Assume $\bigcup_{i=1}^{m} S_i = \mathcal{U}$. $|\mathcal{U}| = n$.

sets have cost $c_1 \dots c_m \geq 0$ (sometimes unit cost is interesting as well).

Solution: sub collection $I \subseteq [m]$

st $\bigcup_{i \in I} S_i = \mathcal{U}$.

Value: cost / cardinality. $\boxed{\text{minimization}}$

wtd S.C. / unweighted S.C.

$\mathrm{cost}(I) = \sum_{i \in I} c_i$ / $|I|$

Algo Greedy : (unweighted)

[while not all elements covered
└ Pick set S_i that covers most uncovered elements.]

Thm : Greedy is a $(\ln n)$-apx algorithm. for unweighted set cover.

Pf : sps $\mathrm{OPT}(I) = K$. then sps n_t = # uncovered elts after t sets.
$n_0 = n$.

Fact : $\exists$ a set that covers $\geq \frac{1}{K}$ frac of current uncovered elts.

Pf : OPT covers the n_t elts. So $\exists$ set in OPT that covers $\geq \frac{n_t}{K}$ elts. ∎

$\Rightarrow n_{t+1} \leq n_t \left(1 - \frac{1}{K}\right)$. $\Rightarrow n_T \leq n_0 \left(1 - \frac{1}{K}\right)^T < n_0 e^{-T/K} = 1$ if $T = K \ln n$. ∎

Fact: Greedy solves the 1-vs-$(1-1/e)$ for Max-k-coverage.

[Def: Max-k-coverage. Solutions = k sets in set system
Val = # elements covered by picked sets.]

Pf: After k picked, $n_t \le n_0 \left(1-\frac{1}{k}\right)^k < n e^{-1}$

$$\Rightarrow \text{coverage fraction} \ge \frac{n(1-1/e)}{n} = 1-1/e. \quad \blacksquare$$

Fact: if $\exists$ algo for Max-k-coverage that is 1-vs-$(1-\delta)$ search algo.
$\Rightarrow \exists\, (\log_{(1/\delta)} n + 1)$ apx algo for min Set Cover.

Pf: "Guess" OPT $= k^*$.

Use algo A. Covers $1-\delta$. repeat on remainder.
with k

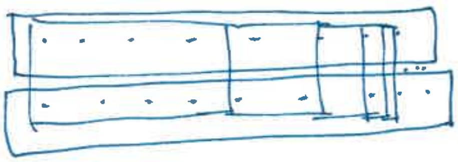
$\Rightarrow$ # uncovered after T rounds $= \delta^T \cdot n < 1$
if $T = \log_{(1/\delta)} n + 1$.

Corollary: 1 vs $(1-1/e)$ for Max k Cov $\Rightarrow$ $\ln n$ for min Set Cover

[redacted]: $\underbrace{\text{if } \ln n(1-\epsilon) \text{ for SC hard}}_{\text{a little more complicated, maybe see.}} \Rightarrow \underbrace{(1-1/e+\delta) \text{ for MaxkCov is hard.}}_{\text{see this in a future lecture.}}$

Algorithmic Gap: Does greedy do better than $\ln n$?

Fact: Greedy no better than $\ln n\,(1-\varepsilon)$. (even for unweighted)

Pf: 

$OPT = 2$.

$Alg = \log_2(n/2)$ $\Rightarrow$ algo gap $= \dfrac{\log_2 n - 1}{2} \cong \dfrac{\ln n}{2 \ln 2}$.

To get $\approx \ln n$, use setting where $OPT = k$ vertically

But each other set covers $1/k$ of remainder.

$$\Rightarrow \#\text{sets} = \log_{\frac{1}{(1-\frac{1}{k})}} n. \quad \Rightarrow \text{gap} = \frac{\log_{(1-1/k)} n}{k} \cong \frac{\ln n}{k \ln\left(\frac{1}{1-\frac{1}{k}}\right)}$$

but $\ln(1+\varepsilon) = \varepsilon \pm \Theta(\varepsilon^2)$ for ε small. $\cong \left(1 - \Theta\left(\frac{1}{k}\right)\right) \cdot \ln n$.

So another algorithm?

Before that: a greedy algo for weighted case.

at each step, pick set that $\max\left(\dfrac{\text{coverage}}{\text{cost}}\right)$.

Thm: Greedy is $\ln n$-apx for weighted set cover.

Pf: (sketch) Same idea as before. Show that if costs $c_1, c_2 \dots c_t$ for sets in steps $1, 2 \dots t$

$$n_t \le n\left(1 - \frac{c_1}{OPT}\right)\left(1 - \frac{c_2}{OPT}\right) \cdots \left(1 - \frac{c_t}{OPT}\right)$$

$$\le n \exp\left(\frac{\sum_i c_i}{OPT}\right).$$

etc. ∎

Linear Program - based Algos:

Idea: Relax-and-Round

1. write an IP for Set Cover. (IP = Integer (Linear) Program).
2. "Relax" it to an LP (LP = Linear Program).
3. Solve this LP. Fact: can solve LPs in poly time.
4. "Round" the fractional solution to Integers.

Usually: ① $IP(I) = Opt(I)$.

② $LP(I) \le IP(I)$.

③ $Alg(I) \le \alpha . LP(I) \Rightarrow Alg(I) \le \alpha . OPT(I)$.

Set Cover: variable $x_S \in \{0,1\}$ for each set $S \in \{S_1, S_2 \ldots S_m\}$.

$\boxed{IP.}$

$$\min \sum c_S x_S$$

$$\text{st} \sum_{S: e \in S} x_S \ge 1 \quad \forall e \in U.$$

$\cancel{x_S \in \{0,1\}.}$

$\boxed{LP}$ $x_S \ge 0$

Round: Imagine each x_S as a prob. value. (Fact: $x_S \in [0,1]$, no reason for x_S to be larger).

Algo: [For $T =$ ______ times

$\forall S \in \mathcal{F}$

select S independently w.p. x_S.]

Towards sampling

What if this is not a feasible solution?

Clean-up: $\forall$ element e, pick cheapest set covering e if e not covered by sampling.

Lemma: $E[\text{cost of solution}] \le T \cdot LP(I) + \left[\sum_{e \in U} (\text{cheapest set covering } e)\right] e^{-T}$.

Pf: $E[\text{cost of each round}] = \sum_S c_S \cdot \Pr[S \text{ picked}] = \sum c_S x_S = LP(I)$.

Lin of exp.

Now: prob(e not covered in one round) $= \prod_{S: e \in S} (1 - x_S) \le e^{-\sum_{S: e\in S} x_S} \le e^{-1}$

$\Rightarrow \Pr(e \text{ not covered in } T \text{ rounds}) \le e^{-T}$.

Now use Linearity of expectation again. ▨

Hence set $T = \ln n$.

$E[\text{cost}] \le (\ln n) \cdot LP + \frac{1}{n} \cdot n \cdot LP$ (b/c LP value $\ge$ cheapest set covering e for any e)

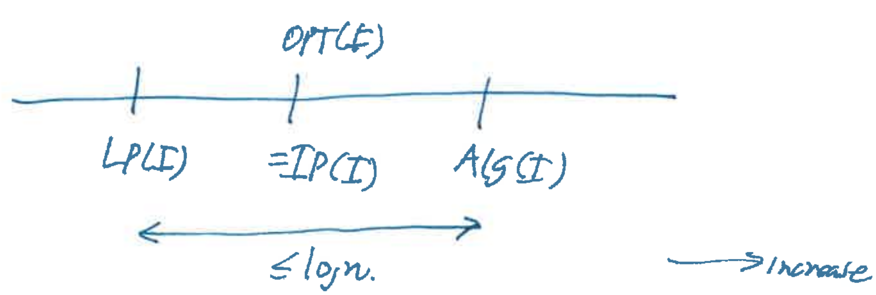
$\boxed{= (\ln n + 1) LP.}$

— x —

HW: Show that if sets are of size B, then LP rounding gives $O(\ln B)$ apx.

Greedy too (but see more later).

Picture



Ask 2 questions:

↗ using this LP+ rounding, cannot beat lg n.

① Algorithmic gap: does ∃ instance where $\frac{Alg(I)}{OPT(I)} = \Omega(\lg n)$.

② Integrality gap: does ∃ instance st $\frac{OPT(I)}{LP(I)} = \Omega(\lg n)$.

→ shows that using this ~~approach~~ LP cannot beat the log-apx.
no matter what rounding we do.
[as long as we relate ourselves to the LP value, of course!].

Algo gap: $\boxed{\text{see in HW}}$.

Integrality gaps:

- Take $U = \{x \in \{0,1\}^d \mid \|x\|_1 = d/2\}$ $\quad d \approx \log_2 n$.

  $n = |U| = \binom{d}{d/2} \approx \Theta\left(\frac{2^d}{\sqrt{d}}\right)$.

- Sets: all "dictator" sets $S_i = \{x \in U \mid x_i = 1\}$. costs $= 1$.

- OPT $\geq d/2 + 1$ Else ∃ element not covered

- LP value: set $2/d$ on each set. (ie. $x_S = 1$ $\forall S$).

  $\Rightarrow$ total LP value $= d \cdot 2/d = 2$.

$\Rightarrow$ Integrality gap $\geq \left(\frac{d/2+1}{2}\right) = \Omega(d) = \Omega(\log n)$.

Fact: Can do better, get $\Omega(\ln n)$ for integrality gap as well.