Lecture 1 - Intro

Friday, December 2, 2022 04:25

① Take aways from this course

- Using Randomness in algorithm design
  - Works well in practice
  - Can often be made deterministic
- Coping with NP-completeness

problem is NP-COMPLETE

⇓

No algorithm solves

ALL INSTANCES — RESTRICTED INPUTS

OPTIMALLY — APPROXIMATION

In Poly time — EXACT EXPONENTIAL TIME

PARAM. ALGORITHMS

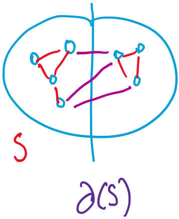
EXAMPLE:

Max Cut

In: G

Task: Find $S \subseteq V(G)$ s.t.

$|\partial(S)|$ is maximized



NP-complete!

Find a solution which is "as good as possible"

umm
can we prove
something?

ALG:

Pick $S \subseteq V(G)$ u.a.r.

Output S

S is a random variable

$\Downarrow$

$|\partial(S)|$ is a random variable

lets compute $E[|\partial(S)|]$

$$E[|\partial(S)|] = \sum_{i=0}^{\infty} i \cdot \underbrace{\Pr[|\partial(S)| = i]}$$

looks hard to compute, right?

[It's actually NP-hard ... why?]

"Method of indicator r.v.s"

Lets try to compute this a different way.

$\forall e \in E(G)$ define $X_e = \begin{cases} 1 & \text{if } e \in \partial(S) \\ 0 & \text{otherwise} \end{cases}$

$$|\partial(S)| = \sum_{e \in E(G)} X_e$$

"Linearity of expectation"

so

$$E[\partial(S)] = \overbrace{E\Big[\sum_{e \in E(G)} X_e\Big] = \sum_{e \in E(G)} E[X_e]}$$

?

? $E[X_e]$

$= 0 \cdot \Pr[X_e = 0] + 1 \cdot \Pr[X_e = 1]$

$= \Pr[X_e = 1]$

$= 1/2$ (why?)

$=$

$= \sum_{e \in E(G)} 1/2 = m/2$

So $E[\partial(S)] > m/2$!

What does this mean?

For every graph G on m edges,

if we sample S_1, S_2, S_3 ......

u.a.r

then $\sum_{i=1}^{t} \frac{|\partial(S_i)|}{t}$ tends to $m/2$

as $t \to \infty$.

✱ Thm: For every random variable X,

- $\Pr[X \geq E[X]] > 0$
- $\Pr[X \leq E[X]] > 0$

Why is this true?

Thm:

$\forall$ graph G, $\exists S \subseteq V(G)$ s.t $|\partial(S)| > m/2$

Pf: $E[|\partial(S)|] = m/2$ + ☆

Ok… but can we find such an S?

Naive: $\Pr[X \geq E[X]] \geq 1/2$

so $\Pr[\,|\partial(S)| \geq E[|\partial(S)|]\,] = 1/2$

Consider a r.v. X that is
0 with prob $1 - 1/1000000$
1 with prob $1/1000000$

$E[X] = 1$, $\Pr[X \geq E[X]] = 1/1000000$

$$\boxed{\Pr[\,|\partial(S)| \geq m/2\,] = ???}$$

MARKOV INEQUALITY

$\forall$ r.v. X s.t $X \geq 0$
$\Pr[X \geq c \cdot E[X]] \leq 1/c$

"no more than half of a population can be at least twice as tall as the average"

Markov: $\Pr[\partial(S) \geq 1 \cdot E[X]] \leq 1$

WRONG DIRECTION

USELESS BOUND

Define $X = m - |\partial(S)|$

Then $E[X] = E[m - |\partial(S)|] = m - E[|\partial(S)|] = m/2$

$$\Pr[|\partial(S)| \geq m/2]$$

$$= 1 - \Pr[|\partial(S)| < m/2]$$

$$= 1 - \Pr[|\partial(S)| \leq m/2 - 1/2]$$

$$= 1 - \Pr[X \geq E[X] + 1/2]$$

$$\geq 1 - \left(1 - \frac{1}{m+1}\right) = \frac{1}{m+1}$$

$$\Pr[X \geq E[X] + 1/2]$$

$$= \Pr\left[X \geq \frac{(E[X] + 1/2)}{E[X]} E[X]\right]$$

$$\underset{\text{MARKOV}}{\leq} \frac{E[X]}{E[X] + 1/2} = \frac{m/2}{m/2 + 1/2} = \frac{m}{m+1} = 1 - \frac{1}{m+1}$$

Dude thats a pretty low success probability 😐

__Idea:__ Repeat t times, pick best soln.

What is $\Pr[\text{best of } t \text{ solns} \geq m/2]$?

$$\Pr[\text{best of } t \text{ solns} \geq m/2]$$

$$= 1 - \Pr[\text{all } t \text{ solns are} < m/2]$$

$$= 1 - \left(1 - \frac{1}{m+1}\right)^t$$

$$\geq 1 - \left(e^{-1/m+1}\right)^t = 1 - e^{-\frac{t}{m+1}}$$

THE INEQUALITY

$1+x \leq e^x$

$1-x \leq e^{-x}$

$\forall\, 0 \leq x \leq 1 \quad 1+x \geq 2^x$

$\forall\, 0 \leq x \leq ½ \quad 1-x \geq 4^{-x}$

✓ $\left(= 1 - 1/e \geq ½ \text{ for } t = m+1 \right)$

$\left(= 1 - 1/e^{100} > 0.9999999999 \text{ for } t = 100(m+1) \right)$

Moral: If we have success probability $1/r$ we need to do $\sim r$ trials to get one success with prob $\sim 1/2$

$$\boxed{\text{TIME} = 1/\text{SUCCESS PROB}}$$

Time per trial: $O(m)$

# trials: $O(m)$

$\Big\}$ total time $O(m^2)$

OBS: $\forall$ graph G

$\text{ALG}(G) \geq m/2 \geq \frac{\text{OPT}(G)}{2}$

↳ ALG is never more than a factor

2 WORSE THAN OPT

↓

ALG is a 2-APPX ALGORITHM.

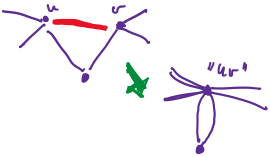
Min Cut

In: G

Task: Find S s.t $|\partial(S)|$ is minimized

$(S \neq \emptyset,\ S \neq V(G))$

CONTRACTION:



ALG:

```
While $|V(G)| > 2$
    pick random edge $uv$
    contract $uv$
```

$V(G)$ is $\{x, y\}$

Output $S =$ all vertices contracted into x

Runtime: polynomial in n, m

Let $c =$ value of min cut.

What is $\Pr[|\partial(S)| \leq c]$??

Let C be some unknown optimal soln

(so $|\partial(C)| = c$)

What is $\Pr[S = C]$??

... probably $\frac{1}{2^n}$ or something??

$$\Pr[S = C] = \Pr[\text{no edge of } \partial(C) \text{ is contracted}]$$

EXAMPLE HERE!

$$\Pr[\text{no edge of } \partial(C) \text{ is contracted}] =$$

$$\frac{m_1 - k}{m_1} \cdot \frac{m_2 - k}{m_2} \cdot \frac{m_3 - k}{m_3} \cdots\cdots \frac{m_{n-2} - k}{m_{n-2}}$$

$$\boxed{\text{OBS: } m \geq \frac{nk}{2} \quad \text{so} \quad m_i \geq \frac{n_i \cdot k}{2}}$$

$$\geq \frac{\frac{n_1 \cdot k}{2} - k}{\frac{n_1 \cdot k}{2}} \cdot \frac{\frac{n_2 \cdot k}{2} - k}{\frac{n_2 \cdot k}{2}} \cdots$$

$$= \frac{(n_1 - 2) \cdot k}{n_1 k} \cdot \frac{(n_2 - 2) k}{n_2 k} \cdots$$

$$= \frac{n_1 - 2}{n_1} \cdot \frac{n_2 - 2}{n_2} \cdot \frac{n_3 - 2}{n_3} \cdots$$

$$= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots\cdots \frac{2}{4} \cdot \frac{1}{3}$$

$$= \frac{1}{n} \cdot \frac{4}{n-1} \cdot \frac{1}{1} \cdot \frac{1}{1} \cdots \cdot \frac{2}{1} \cdot \frac{1}{1} = \frac{2}{n(n-1)}$$

For every optimal cut C

Pr algorithm outputs $S = C$ is $\geq \frac{2}{n^2}$!!

(repeat $O(n^2)$ times for success with prob $\sim 1/2$)

Thm: For every graph G, the number of min cuts is $\leq \frac{n \cdot (n-1)}{2}$

Pf: ?!!

Min k-Cut

In: G, k

TASK: FIND PARTITION of $V(G)$ into non-emty sets $S_1, S_2, \ldots S_k$

minimizing $\left| \bigcup_{i \neq j} \partial(S_i, S_j) \right|$

Min k-cut is NP-complete

[Reduce from CLIQUE in <u>regular</u> graphs]

Karger et al:

For every graph G, integer k,
optimal k-cut $C_1, C_2 \dots C_k$
$\Pr[\text{alg outputs } C_1, C_2 \dots C_k] \geq 1/n^{2k}$

Gives n^{2k}-time algorithm for min k-cut.

poly time for every k.

why didn't we show $P = NP$?

XP
SLICEWISE POLYNOMIAL

Exponent gets worse with k.

CAN WE GET

$$\left.\begin{array}{l} 2^k \cdot n^3 \quad \dots \\ \text{or} \quad 2^{2^{2^{2^k}}} \quad n^{10000} \end{array}\right\} \text{FIXED PARAMETER TRACTABLE}$$

Nope... would imply
an FPT algorithm for k-CLIQUE
which implies FPT = W[1]

CLIQUE: In: G

- TASK: Find $S \subseteq V(G)$ s.t. $|S|$ is maximized
  and $\forall\, u,v \in S$ $uv \in E(G)$

- Known: NP-c.
  No good appx alg unless P=NP
  factor $n^{1-\varepsilon}$

naive $k^2 n^k$
clever $n^{\frac{\omega k}{3} + o(1)}$

no $f(k) \cdot n^{o(k)}$ time alg. unless
Exponential Time Hypothesis (ETH)
fails

- What if input is a random graph?
  $G(n,p)$ model:
  - n vertices (labeled $1 \dots n$)
  - $\forall$ pair u,v of vertices $uv \in E(G)$
    with probability p.

ALG FOR MAX CLIQUE IN $G(n, 1/2)$ model:

```
$S = v_1$
For $i = 2$ to $n$
    if $N(v_i) \supseteq S$ then
        $S = S \cup \{v_i\}$
```

Output S

- S is a clique
- How big is S?
  - In step i, $\Pr[N(v_i) \supseteq S] = 1/2^{|S|}$
  - Let T_i be the random variable denoting # vertices alg considers from i'th vertex is added to S until $i+1$'st vertex is.

$$E[T_i] = 1 + \frac{1}{2^i} \cdot 0 + \left(1 - \frac{1}{2^i}\right) E[T_i]$$

$$E[T_i] = 2^i$$

Let $X_t = T_1 + T_2 + T_3 \dots T_t$ = time until $|S| = t$

Then $E[X] = \sum_{i=1}^{t} 2^i = 2^{t+1} - 1 \leq 2^{t+1}$