6.842 lecture 2:

The Lovász Local Lemma

The Lovász Local Lemma

Another way to argue that it's possible that "nothing bad happens"

If $A_1 A_2 \ldots A_n$ are <u>bad</u> <u>events</u>

how do we know that there is a positive probability that

<u>none</u> occur?

$Pr[A_i] \neq 1 \ \forall i$

if A_i's independent + "nontrivial":

$$Pr[\cup A_i] \leq 1 - Pr[\cap \bar{A}_i]$$
$$= 1 - \prod \underbrace{Pr(\bar{A}_i)}_{>0}$$
$$< 1$$

else, usual way: Union Bound

no assumptions on A_i's with respect to independence

$$\Pr[\cup A_i] \leq \sum \Pr[A_i]$$

if each A_i occurs with prob $\leq p$,
then need $p < \frac{1}{n}$ to get
interesting bound i.e. $\Pr[\cup A_i] < 1$

What if A_i's have "some" independence?

<u>def.</u> A "independent" of $B_1, B_2, \ldots B_k$ if

$\forall J \subseteq [k]$, $J \neq \emptyset$ then $\Pr[A \cap \bigcap_{j \in J} B_j]$
$= \Pr[A] \cdot \Pr[\bigcap_{j \in J} B_j]$

note: $[k]$ means $\{1 \ldots k\}$

<u>def.</u> $A_1 \ldots A_n$ events
$D = (V, E)$ with $V = [n]$ is
"dependency digraph of $A_1 \ldots A_n$"
if each A_i independent of all A_j that
are not neighbors in D (i.e. all A_j s.t. $(i,j) \notin E$)

Lovász Local Lemma (symmetric version)

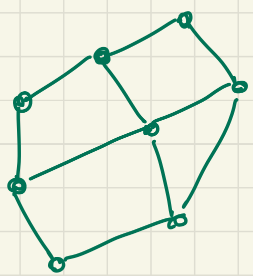
$A_1 \dots A_n$ events s.t. $\Pr(A_i) \le p \;\; \forall i$

with dependency digraph D s.t. D

has max degree $\le d$.

If $ep(d+1) \le 1$ then

$$\Pr\left[\bigcap_{i=1}^{n} \bar{A}_i\right] > 0$$



Union bnd

need

$p < \frac{1}{n}$

LLL

if $d \le 4$

only need

$p \le \frac{1}{e \cdot (4+1)}$

Application

Thm. Given $S_1 \dots S_m \subseteq X$ $\quad |S_i| = l$

each S_i intersects at most d other S_j's

previously needed $m < 2^{l-1}$

now no restriction on m

but there is a restriction on "degree"

if $e \cdot (d+1) \leq 2^{l-1}$

then can 2-color X such that

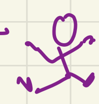
each S_i not monochromatic

ie. $\mathcal{H}$ is hypergraph with m edges each containing l nodes + each intersecting $\leq d$ other edges

Pf color each elt of X red/blue iid with prob $\frac{1}{2}$

A_i = event that S_i is monochromatic

even the proof starts out the same!



$p = \Pr[A_i] = 1/2^{l-1}$

A_i indep of all A_j s.t. $S_i \cap S_j = \emptyset$

so depends on $\leq d$ other A_j

since $e \cdot p \cdot (d+1) = e \cdot \frac{1}{2^{\ell-1}} \cdot d+1 \le 1$ by assumption

LLL $\Rightarrow \exists$ 2-coloring $\blacksquare$

Comparison:

#edges $= m$

size of edges $= \ell$

$m < 2^{\ell-1}$

#edges $= m$

size of edges $= \ell$

each edge intersects with $\le d$ others

$d+1 \le \frac{2^{\ell}}{e}$

no dependence on m

Application 2: Boolean Formulae

Given CNF formula s.t. ℓ vars in each clause + each var in $\leq k$ clauses

If $\frac{e(\ell k+1)}{2^{\ell-1}} \leq 1$ there is a satisfying assignment.

How do you find a solution?

partial history:

Lovász	1975	nonconstructive (no fast algorithm to find soln)	$d \leq 2^{\ell-1}$
Beck	1991	randomized algorithm but for more restrictive conditions on parameters	$d \leq 2^{\ell/1000}$
Alon	1991	parallel version	$d \leq 2^{\ell/8}$
	⋮		⋮

Moser 2009 negligible restrictions for SAT + most other problems $d \le \frac{2^{l-1}}{c}$

Moser Tardos

⋮

Moser-Tardos Thm:

Given $S_1 \dots S_m \subseteq X$, s.t. each S_i intersects $\le d$ other S_j's.

If $e(d+1) \cdot c \le 2^{l-1}$ then can find 2-coloring of X s.t. each S_i not monochromatic in time poly in $m, d, |X|$.

Moser – Tardos Algorithm

(1) 2-color all elements of X randomly
(pr $\frac{1}{2}$, iid)

(2) while not proper 2-coloring of S_i's
- pick (arbitrary) monochromatic S_i + randomly reassign colors to elements of S_i

we will do Beck-like algorithm, (stronger assumptions, much slower, more complicated algorithm, hopefully easier to explain?)

Stronger assumptions:

(1) For today, assume ℓ, d constants

(2) Binary Entropy: $H(x) \equiv -x\log_2 x - (1-x)\log_2(1-x)$

Let $p = 2 \cdot 2^{(H(\alpha)-1)\cdot \ell}$

$$e d p^{\frac{1}{d+1}} < \frac{1}{2}$$

(3) $2e(d+1) < 2^{\alpha n}$

Algorithm: Given $S_1 \ldots S_m \subseteq X$ $\quad |S_i| = \ell \ \forall i$

First pass:

for each $j \in X$ pick color red/blue via coin toss

S_i is "bad" if $\leq \alpha \cdot \ell$ reds
<u>or</u> $\leq \alpha \cdot \ell$ blues

$B \leftarrow \{ S_i \mid S_i \text{ is bad} \}$

1st pass is successful if all "connected components" of B are $\leq d \log m$

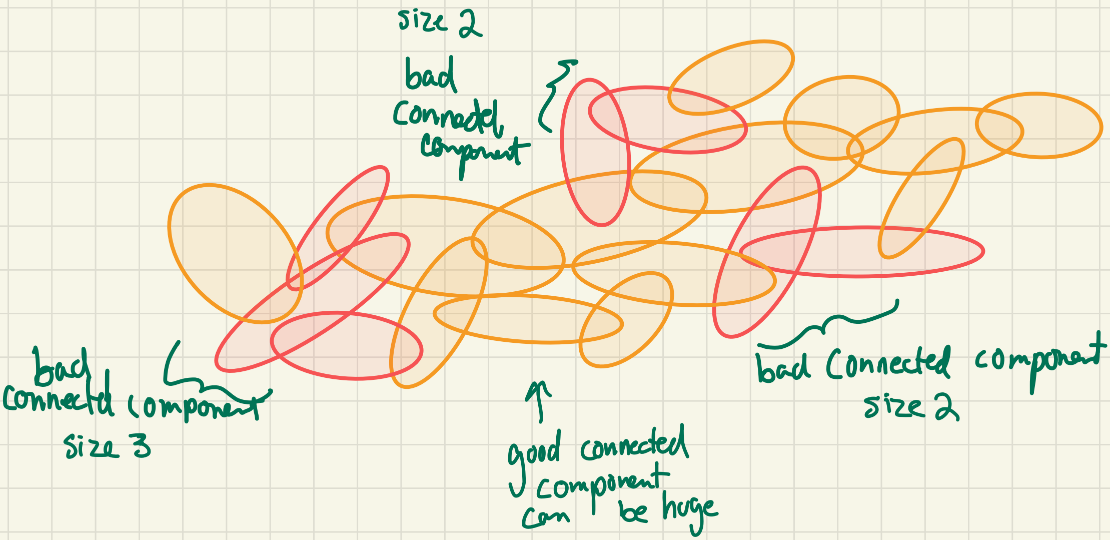
edge bet A_i & A_j if $A_i \cap A_j \neq \emptyset$

(if not succesful, retry)

Second Pass:

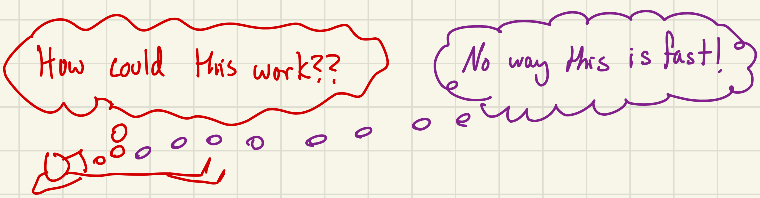
Brute force each connected component (w/o violating their nbrs)

few sets so maybe efficient?



Some questions:

- why is output legal? what if changing S_i's in B makes S_j & B monochromatic?
- How many times to repeat pass 1?
- How fast is pass 2?



Why is output legal?

First pass:

for each $j \in X$ pick color red/blue via coin toss

S_i is "bad" if $\le \alpha \cdot \ell$ reds
or $\le \alpha \cdot \ell$ blues

$B \leftarrow \{S_i \mid S_i \text{ is bad}\}$

pass successful if all "connected components" of bad S_i's are $\le d \log m$

(if not succesful, retry)

Second Pass: Brute force each connected component

If S_i not bad + $< \alpha n$ nodes in bad nbrs

then S_i will still be bichromatic after recoloring.

If S_i bad + has $\ge \alpha \ell$ nodes in bad nbrs,

then $\ge \alpha \ell$ nodes get recolored

- if recolored randomly, $\Pr[S_i \text{ is monochrom}] < 2^{-\alpha \ell}$
- using LLL

assumption ✱ + assume $2e(d+1) < 2^{\alpha \ell}$

$\Longrightarrow$ solution exists!

How many repetitions of Pass 1?

fact for $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$

$$\forall S_i, \quad \Pr[S_i \text{ bad}] \le 2 \cdot \sum_{i \le \alpha n} \binom{\ell}{i} / 2^\ell \le 2 \cdot \underbrace{2^{(H(\alpha)-1)\ell}}_{\text{define this to be } p}$$

$\le p$

$\approx 2^{-c\ell}$ for some const c

Given dependency digraph G,

put edge between S_i & S_j if $S_i \cap S_j \neq \emptyset$

if $S_{i_1}, S_{i_2}, \dots, S_{i_m}$ are independent set

no edges between them

(so $S_{i_k} \cap S_{i_\ell} = \emptyset \;\; \forall\, i_k, i_\ell$)

then $\Pr[S_{i_1} \dots S_{i_m} \text{ all in } B] \le p^m$

since mutually independent

First try

Show no big component survives:

$$\Pr[\text{specific } \underset{\uparrow \text{ size } s}{\text{big}} \text{ component survives}]$$

$$\leq \Pr[\underset{\uparrow \text{ size } s' < s}{\text{big}} \text{ independent set in component survives}]$$

$$\leq p^{s'}$$

$$\Pr[\text{any big component survives}]$$

$$\leq \underbrace{\# \text{ big components}}_{} \cdot \underbrace{p^{s'}}_{}$$

what is a good bound?

$\binom{n}{s}$? way too big!!

Can use degree bound to improve!!

how does s' compare to s?

if component is clique, then s' could be 1

but, use degree bound!

Plan: hope to show no big component survives.

if big component C survives,

can get good bound on # bounded degree subtrees! $\longrightarrow$ then C has a big subtree that survives

then can find (less) big independent

since bounded degree $\longrightarrow$ set in subtree

Well Known fact:

subtrees of size u in graph of degree $\leq \Delta$ is $\leq n \cdot \frac{1}{(\Delta-1)(u+1)} \binom{\Delta u}{u}$

#nodes $= n$

$$\leq \underbrace{n(e\Delta)^u}$$

much much better than $\binom{n}{u}$ when Δ is constant

Given subtree of size u,

it has indep set of size $\geq \dfrac{u}{\Delta+1}$

why?

each round:
- I gets bigger by 1
- subtree gets smaller by $\leq \Delta+1$

Repeat
$I \leftarrow$ arbitrary node u in subtree
remove u + all nbrs of u from subtree
Until subtree is empty

$\Rightarrow$ # rounds $= |I| \geq \dfrac{u}{\Delta+1}$

New try:

Show no big component survives:

$$E[\#\text{ of size} > S \text{ subtrees that survive}]$$

$$\leq \sum_{i=s}^{m} E[\#\text{ size } i \text{ subtrees that survive}]$$

hiding an indicator argument in there :)

$$\leq \sum_{i=s}^{m} (\#\text{ size } i \text{ subtrees}) \times \Pr[\text{size } i \text{ subtree survives}]$$

$$\leq \sum_{i=s}^{m} m \cdot \underbrace{(e\,d)^i \times \left(p^{\frac{i}{d+1}}\right)}_{\left(e\,d\,p^{d+1}\right)^i}$$

(*) assume this is $< 1/2$

$$\leq \sum_{i=s}^{m} m \cdot \frac{1}{2^i} < \frac{m}{2^{s-1}}$$

for $s = \log 4m$

$$\leq \frac{m}{4m} = 1/4$$

upper bound on expected # of big components

By Markov's ≠:

i.e. ≥ 1

$\Pr[\#$ of size $\geq \log 4m$ subtrees $> 0] < \frac{1}{4}$

so $\Pr[\#$ components of size $\geq \log 4m$ is $> 0] < 1/4$

$\Longrightarrow$ expected # times to repeat first pass ≤ 4

How fast is Pass 2?

\# Surviving components $\leq O(\log m)$

\# settings to vars in surviving components $\leq 2^{\ell \, O(\log m)}$

$$= m^{O(\ell)}$$

if ℓ is constant: poly(m) time * assumption

else, recurse on components