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 bad events, how do we know that there is a positive probability that none occur?

if $A_i$'s independent & "nontrivial":

$$\Pr[\cup A_i] \leq 1 - \Pr[\bigwedge \overline{A_i}]$$

$$= 1 - \prod \Pr(\overline{A_i})$$

$$< 1$$
else, usual way: \( \text{Union Bound} \)

\[
\Pr[\bigcup A_i] \leq \sum \Pr[A_i]
\]

if each \( A_i \) occurs with prob \( \leq p \),

then need \( p < \frac{1}{n} \) to get

interesting bound ie, \( \Pr[\bigcup A_i] < 1 \)

What if \( A_i \)'s have "some" independence?

def. \( A \) "independent" of \( B_1, B_2 \ldots B_k \) if

\[
\forall \ J \subseteq [K] \quad \text{then} \quad \Pr[A \cap \bigcap_{j \in J} B_j] = \Pr[A] \cdot \Pr[\bigcap_{j \notin J} B_j]
\]

Note: \([K]\) means \([1 \ldots K]\)

def. \( A_1 \ldots A_n \) 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 \( D \) (ie. all \( A_j \) st. \((ij) \in E\) )
Lovász Local Lemma (symmetric version)

$A_1, \ldots, A_n$ events s.t. $\Pr(A_i) \leq p \forall i$ with dependency digraph $D$ s.t. $D$ has max degree $\leq d$.

If $\Pr(d+1) \leq 1$ then

$$\Pr[\bigwedge_{i=1}^n \overline{A_i}] > 0$$
**Application**

**Thm.** Given $S_1 \ldots S_m \subseteq X$, $|S_i| = l$ 

- Each $S_i$ intersects at most $d$ other $S_j$'s 
- If $e(d+1) \leq 2^{l-1}$ 
- Then can 2-color $X$ such that 
  - Each $S_i$ not monochromatic
  
  i.e. $H$ 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 $p = \frac{1}{2}$
- $A_i$ event that $S_i$ is monochromatic

$$P = \Pr[A_i] = \frac{1}{2}^{l-1}$$

- $A_i$ indep of all $A_j$ s.t. $S_i \cap S_j = \emptyset$
- So depends on $\leq d$ other $A_j$
since \( e^p (d+1) = e \cdot \frac{1}{2^{d+1}} \cdot d+1 \leq 1 \)

\[ \text{LLL} \Rightarrow \exists \text{ 2-coloring} \quad \bullet \]

Comparison:

- #edges = \( m \)
- size of edges = \( l \)
  - \( m \leq 2^{l-1} \)

Each edge intersects with \( \leq d \) others

- #edges = \( m \)
- size of edges = \( l \)
  - no dependence on \( m \)
  - \( d+1 \leq \frac{2^l}{e} \)
Application 2: Boolean Formulae

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

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

How do you find a solution?

partial history:

Lovász 1975 nonconstructive (no fast algorithm to find soln)

Beck 1991 randomized algorithm but for more restrictive conditions on parameters

Alon 1991 parallel version

$\frac{d}{2^{\theta}}$
Moser 2009 negligible restrictions for SAT & most other problems

c \leq \frac{2^{l+1}}{c}

Moser Tardos

Moser-Tardos Thm:

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

If $e(d+1) \cdot c \leq 2^{l-1}$ 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 
(precisely $\frac{1}{2}$, 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. \( \text{Binary Entropy} \): \( H(x) \equiv -x \log_2 x - (1-x) \log_2 (1-x) \)

   Let \( p = 2 \cdot 2^{(H(x)-1) \cdot l} \)

   \[ ed \ p \frac{1}{d+1} \leq \frac{1}{2} \]

3. \( 2e(d+1) \leq 2^an \)
Algorithm: Given $S_1...S_m \subseteq X$ \quad $|S_i|=l \ \forall i$

First pass:

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

$S_i$ is "bad" if \quad $\leq \alpha \cdot l$ reds \quad or \quad $\leq \alpha \cdot l$ blues

$B = \{ S_i | S_i \text{ is bad} \}$

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

(If not successful, retry)

Second Pass:

Brute force each connected component

(w/0 violating their nbrs)
Some questions:

- Why is output legal? What if changing $S_i$'s in $B$ makes $S$ dB monochromatic?
- How many time to repeat pass 1?
- How fast is pass 2?

How could this work??

No way this is fast!
Why is output legal?

First pass:

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

$S_i$ is "bad" if $\leq \alpha d$ reds or $\leq \alpha d$ blues

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

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

(if not successful, retry)

Second Pass: Brute force each connected component

If $S_i$ not bad and $< \alpha d$ nodes in bad nbrs

then $S_i$ will still be bichromatic after recoloring.

If $S_i$ bad and has $\geq \alpha d$ nodes in bad nbrs,

then $\geq \alpha d$ nodes get recolored.

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

- using LLL

\[ \text{assumption} \quad + \text{ assume } 2e(\alpha d + 1) < 2^\alpha \]

$\Rightarrow$ solution exists!
How many repetitions of Pass 1?

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

$$\forall S, \quad \Pr [S \text{ bad}] \leq 2 \cdot \sum_{i \leq \log n} (\frac{i}{2})^l \leq 2 \cdot \frac{H(n) - 1}{l} \leq \rho$$

define this to be $\rho \approx 2^{-cl}$ for some constant $c$

Given dependency digraph $G$,

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

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

(so $S_{i_k} \cap S_{i_l} = \emptyset \forall i_k, i_l$)

then $\Pr [S_{i_1} \ldots S_{i_m} \text{ all in } B] \leq \rho^m$

since mutually independent

no edges between them
First try

Show no big component survives:

\[ \Pr[\text{specific big component survives}] \]
\[ \leq \Pr[\text{big independent set in component survives}] \]
\[ \leq p^{s'} \]

\[ \Pr[\text{any big component survives}] \]
\[ \leq \# \text{ big components} \cdot p^{s'} \]

What is a good bound?

\[ (s) \] is 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 \[ \Rightarrow \] then C has a big subtree
that survives
then can find (less) big independent
subtree!

Well known fact:

\[ \text{\# subtrees of size } u \text{ in graph of degree } \Delta \text{ is } \leq n \cdot \frac{1}{(\Delta-1)(u+1)} \binom{\Delta u}{u} \]
\[ \text{\# nodes } = n \]

\[ \leq n(e \Delta)^u \]

much much better than \( \binom{n}{u} \)
when \( \Delta \) is constant
Given subtree of size $u$, it has independent set of size $\geq \frac{u}{\Delta+1}$.

Why?

Repeat each round $i$:

- $I$ gets bigger by 1
- Subtree gets smaller by $\leq \Delta+1$

Until subtree is empty

$\Rightarrow$ # rounds $= |I| = \frac{u}{\Delta+1}$
New try:

Show no big component survives:

\[
E\left[\text{\# of size } S \text{ subtrees that survive}\right] \leq \sum_{i=s}^{m} E\left[\text{\# size } i \text{ subtrees that survive}\right] \\
\leq \sum_{i=s}^{m} (\text{\# size } i \text{ subtrees}) \times \Pr\{\text{size } i \text{ subtree survives}\} \\
\leq \sum_{i=s}^{m} m \cdot (ed)^i \times \left(\frac{i}{d+1}\right) \\
\leq \sum_{i=s}^{m} \left(e \cdot d \cdot p \cdot d+1\right)^i \\
\text{assume this is } \leq \frac{1}{2} \\
\leq \sum_{i=s}^{m} \frac{m - \frac{1}{2^i}}{2} \leq \frac{m}{2^{s-1}} \\
\text{for } s = \log 4m \\
\leq \frac{m}{4m} = \frac{1}{4}
\]
By Markov’s \( \implies \):

\[
\text{Pr}[\text{size } = \log_4 m \text{ subtrees } > 0] < \frac{1}{4}
\]

\[
\text{So } \text{Pr}[\text{components } \geq \log_4 m \text{ is } > 0] < \frac{1}{4}
\]

\[
\Rightarrow \text{ expected } \# \text{ time to repeat first pass } \leq 4
\]
How fast is Pass 2?

# surviving components \leq O(\log m)

# settings to vars in surviving components \leq 2^{\log m} = m

if \( l \) is constant: \( \text{poly}(m) \) time * assumption
else, recurse on components