Lecture 2

LLL ① '17

The Lovász Local Lemma

Another way to argue that "nothing bad happens"

If $A_1 \dots A_n$ are bad events

how do we know if there is positive probability that none occur?

usual way: Union bnd

no assumptions on A_i's w.r.t. independence

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

if each A_i occurs with prob p, then need $p < \frac{1}{n}$ to get anything interesting (i.e. sum < 1)

if A_i's independent & "nontrivial":

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

What if A_i's have "some" independence?

def A "independent" of $B_1 \dots B_k$ if $\forall J \subseteq [k]$, $J \ne \emptyset$

$$\Pr[A \cap \bigcap_{j \in J} B_j] = \Pr[A] \cdot \Pr[\bigcap_{j\in J} B_j]$$

<u>def.</u> $A_1 \cdots A_n$ 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 don't neighbor it in D (ie., all A_j s.t. $(i,j) \notin E$)

<u>Lovász Local Lemma</u> (symmetric version)

$A_1 \ldots A_n$ events s.t. $pr(A_i) \leq p$ $\forall i$

with dependency digraph D s.t. D is of degree $\leq d$.

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

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

Application:

new: degree bound restri

<u>Thm</u>. $S_1 \ldots S_m \subseteq S$, $|S_i| = \ell$, each S_i intersects at most d other S_j's

before $m < 2^{\ell-1}$
now m not restricted

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

then can 2-color S s.t. each S_i not monochromatic

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

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

$A_i \equiv$ event that S_i monochromatic

$Pr[A_i] = 2^{-(\ell-1)}$

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

depends on $\leq d$ other A_j

Since $ep(d+1) = e \frac{1}{2^{\ell-1}}(d+1) \leq 1$

LLL $\Rightarrow \exists$ 2-coloring ■

Comparison:

edges $= m$

size of edge $= \ell$

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

edges $= m$

size of edge $\geq \ell$

each edge intersects $\leq d$ others

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

no dependence on m

A second application:

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

LLL algorithm (

How do you find a solution?

partial history:

Lovász 1975 non-constructive (no fast algorithm to find soln) $d \le 2^{\ell \cdot}$

Beck 1991 randomized algorithm but for more restrictive conditions on parameters $d \le 2^{\ell/}$

$d \le 2^{\ell/8}$ ... $d \le$

⋮

Moser 2009 negligible restrictions for SAT $d \le 2$

" " " most problems

Moser Tardos

⋮

Thm given $S_1 \ldots S_m \subseteq \mathcal{S}$

each S_i intersects $\le d$ other S_j's

if $e(d+1) \cdot c \le 2^{\ell-1}$

then can find 2-coloring of $\mathcal{S}$ s.t.

each S_i not monochromatic

in time poly in m, d

Algorithm

- 2-color all elts of S randomly (iid, uniform)
- While there is a monochromatic set:
  - pick arbitrary "violated" S_i
  - randomly reassign colors to elements of S_i

for example run, see p.3

Correctness trivial ✓

Runtime how many recolorings? * see 2a

To analyze, define "witness tree" to explain why a certain event happened.

def. "log of execution" is a set of pairs $(1, S_{i_1})(2, S_{i_2})\ldots$ where first entry is a "loop" number and second entry S_{i_j} is the set resampled at j^{th} loop.

e.g. $(1, S_1)\ (2, S_2)\ (3, S_1)\ (4, S_5)\ (5, S_2)\ldots$

How many recolorings?

what independence properties do we have?

if $S_i \cap S_j = \phi$ then whether they are monochromatic is independent at all times

if $S_i \cap S_j \neq \phi$ but,

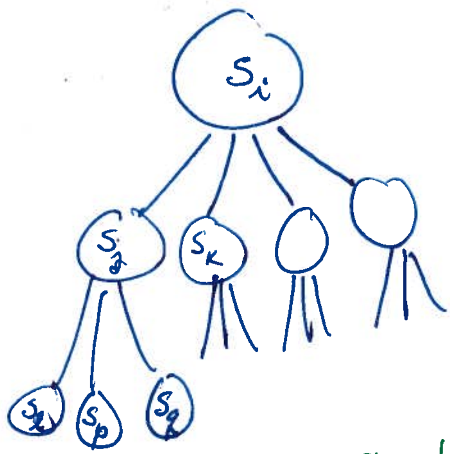
consider: $\Pr[S_i \text{ 2-colored at time } t]$

& $\Pr[S_j \text{ 2-colored at time } u]$

such that there was a recoloring of $S_i \cap S_j$ at time $t \leq v < u$

then also independent!

Model as tree:



← all S_j's s.t. $S_i \cap S_j \neq \phi$

all S_l's s.t. $S_j \cap S_l \neq \phi$

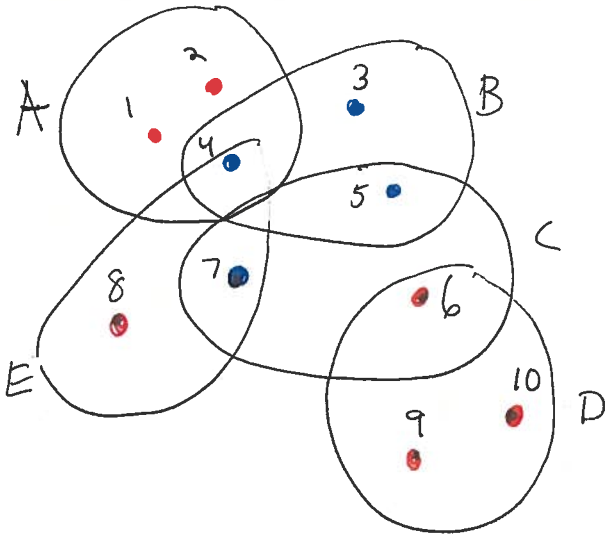
Where is the gain? This tree is d-ary, not n-ary

recolorings of S_i ⟺ recolorings of connected component in this tree

LLL-alg ③

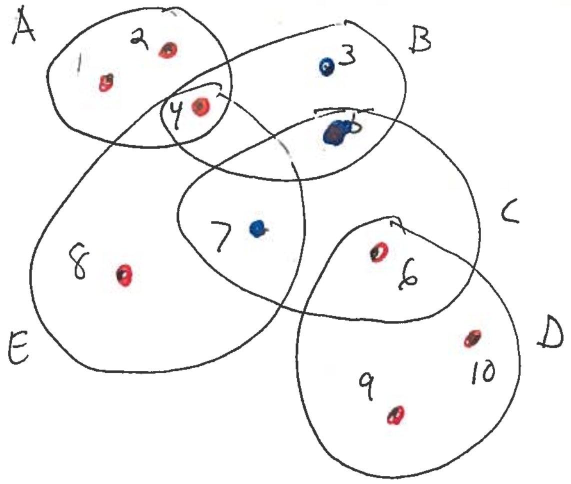
Log: (1,B) (2,D) (3,A) (4,C) (5,E)

example time 0



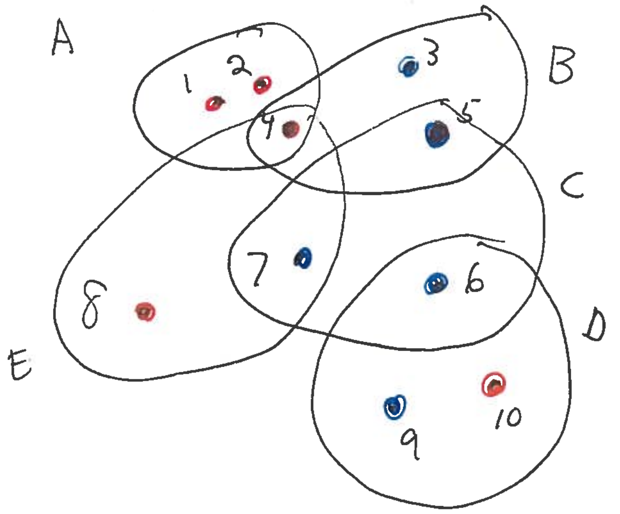
time 1

(1,B) resample edge B



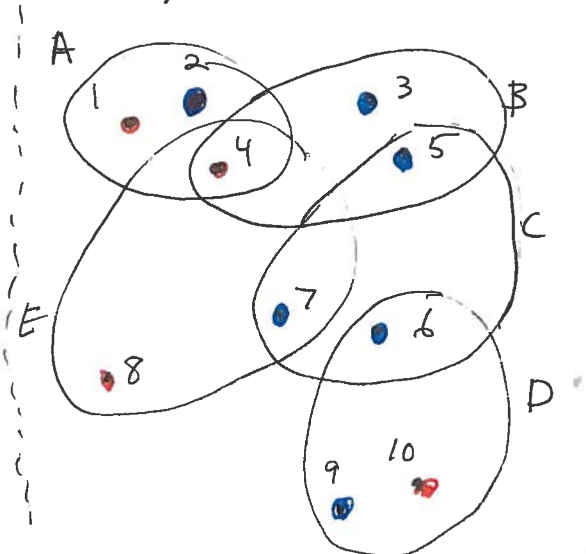
time 2

(2,D) resample D

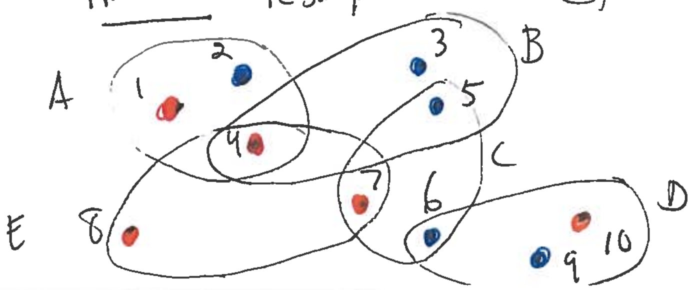


time 3

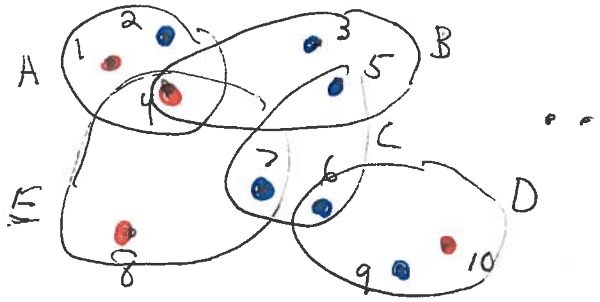
(3,A) resample A



time 4 resample C (4,C)



time 5 resample E (5,E)



A plan :

Show that for any $\underset{\substack{\uparrow \\ \text{lots of resamplings}}}{\text{"long"}}$ log, it is unlikely to happen.

then

$$\Pr[\text{any long log occurs}] \le \underbrace{(\#\text{ long logs})(\max}_{\text{union bnd}} \text{ prob of a long log}) \;?$$

but need to be a bit more elaborate, may be show:

$$\Pr[\text{a log longer than size } t_0]$$

$$\le \sum_{t > t_0} \underbrace{(\#\text{ logs of length } t)}_{\substack{\text{still too many} \\ \text{of these to do naively}}} (\text{Prob of log of length } t)$$

Plan here :

Focus on point of view of each set S_i

+ how labellings can evolve

- not too many ways due to locality
- each big one has low probability

<u>def.</u> "witness tree for step j" $(j>0)$

is constructed as follows

- root vertex labelled by S_{i_j}
- { go backwards thru log }
  - Do for step $j, j-1, j-2, \ldots$
    - if edge relabeled at current step t shares any nodes with edges already in witness tree,

      add S_{i_t} to witness tree by making it point to arbitrary node on witness tree which is at max distance from root

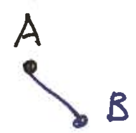
any S_{i_t} can be added many times to witness tree →

In our example:

witness tree for time 1:

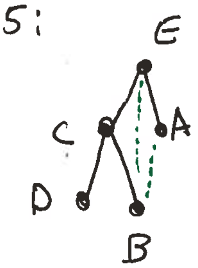
w.t. for time 2:

w.t. for time 3:

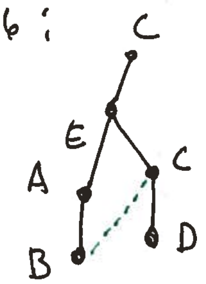


time 4:

time 5:



time 6:



How do we bound probability of specific witness tree Υ in a run?

To analyze prob of tree Υ, upper bound via "Υ-check" procedure

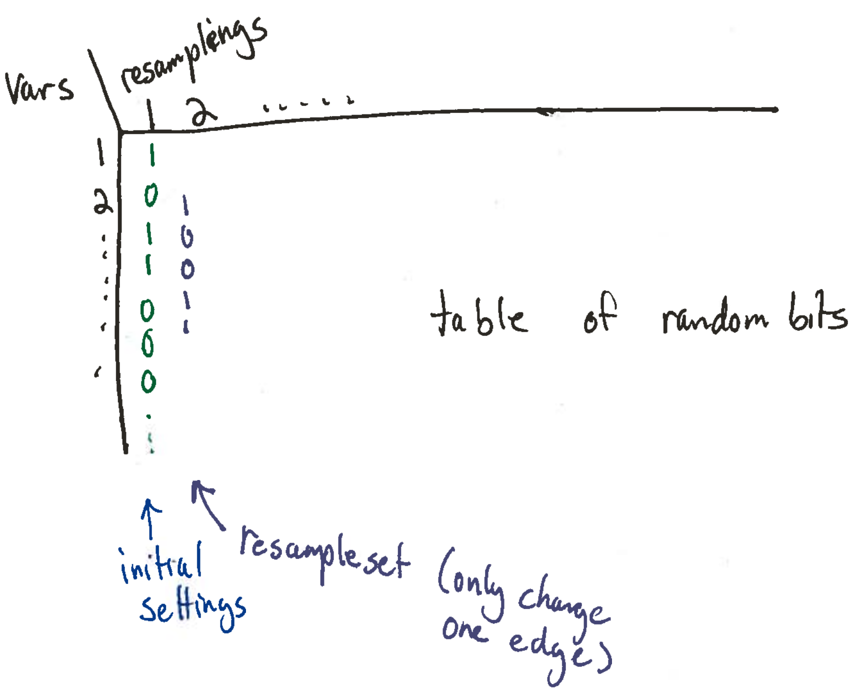
i.e., ensure that:

prob Υ occurs as a witness tree

$\leq$ prob Υ-check passes

Def. Υ-check procedure:

- Visit nodes of Υ in reverse BFS order (max depth first)
- take random evaluations of vars in current set
- check that set is monochromatic (violated)
- pass if all checks are violated



Important point

prob of violation $= 2^{-(\ell-1)} \equiv p$

Observe:

- if 2 sets at same level in tree, can't intersect! (by construction) ⇒ independent (i.e. order of coin tosses doesn't matter)
- if 2 sets at different levels, will resample + get totally new bits → before later set ⇒ independent

note that we consider reverse BFS

$\therefore$ $\Pr[\tau\text{-check passes}] \le p^{|\tau|} = \left(2^{-(\ell-1)}\right)^{|\tau|}$

How to use the τ-check?

1) Prob of getting tree somewhere in log $\le$ prob of τ-check passing

generosity #1

many τ-trees consistent with log. We are bounding prob of any of them. (i.e. sum)

2) no tree occurs twice in log (has to have previous tree as subtree!)

3) so, expected length of log = expected # of distinct trees in log

generosity #2: some of distinct trees can't even happen in our input, we are union bounding over a lot more ~~than~~ can happen

Expected # of resamplings

$$E[\#\text{ resamples}] \le \sum_{\text{roots } i} \sum_{\substack{T \\ \text{with root } i}} E[\#\text{ times labelled tree } T \text{ rooted at } i \text{ occurs in execution of an algorithm}]$$

$$= \sum_{\text{roots } i} \sum_{\substack{T \text{ rooted} \\ \text{at } i}} E[X_T] \quad \text{where } X_T = \begin{cases} 1 & \text{if } T \text{ occurs in} \\ 0 & \text{o.w.} \end{cases}$$

(here we use that no tree occurs twice in a log)

$$= m \sum_{s=1}^{\infty} \sum_{\substack{T : |T| = s \\ T \text{ with fixed root}}} E[X_T]$$

$$\le m \sum_{s=1}^{\infty} \binom{sd}{s-1} p^s \qquad (*)$$

$$\le m \sum_{s=1}^{\infty} \underbrace{((d+1)e)^s p^s}$$

since $p < \frac{(1-\varepsilon)}{e(d+1)}$ this is geometric sum + is $\Theta(1)$

if $p < \frac{1}{2e(d+1)}$ then goes down exponentially + gives good concentrat

$\therefore E[\text{run time}] = E[\#\text{ resamples}] \times \text{time per resample}$

is $\text{poly}(m, \ell, d)$

(*) Why?

How many d-ary labelled rooted trees of size s?

describe via Eulerian tour (left → right):

write 1 if go down
0 if skip child
(2 for "pop up" is redundant)

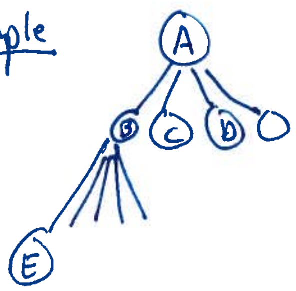
then, each node contributes d bits

String is $\le sd$ characters with $s-1$ 1's

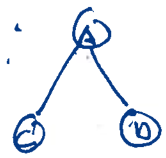
$\le \binom{sd}{s-1}$ such strings

$\le \left(e\,\frac{sd}{s-1}\right)^{s-1} \approx (ed)^{s-1}$ by Stirling's approx

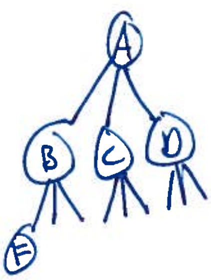
example



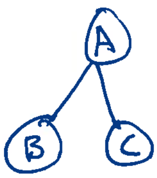
← A's d children



0, 1000, 1000

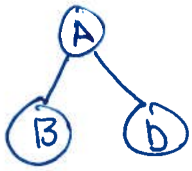


$s = 3$
⟹



1 000 1000, 0

go to B
skip B's kids
go to C
skip C's kids
skip D



1000, 0, 1000

• • •

1, 1000, 00, 00

go to B
go to F
skip F's kids
skip other B's kids
skip other A kids