Lecture 5: The Lovasz Local Lemma (Beyond the Union Bound).

<u>Until now</u>: sps. want to show something good happens. → nothing bad.

<u>Show</u>: $\sum_{i} Pr(\text{bad event}_i) < 1 \Rightarrow Pr(\bigcup_i \text{Bad event}_i) < 1$

i: all bad events

union bound.

$\Rightarrow$:) $Pr(\text{nothing bad happens}) > 0$.

Eg: k-satisfiability.

$\varphi = (x_1 \vee \bar{x}_5 \vee x_9) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_{11}) \wedge \ldots\ldots$ $k=3$.

m clauses, n variables, k-CNF formula

Want to understand when φ <u>must</u> have satisfying assignment.

Pick a random assignment: $X \in \{T, F\}^n$ unif. at random.

B_i = event that clause i is unsatisfied.

$Pr(B_i) = 2^{-k}$ (all vars set "wrong").

$\Rightarrow$ if $m \cdot 2^{-k} < 1 \Rightarrow Pr(\bigcup_i B_i) < 1 \Rightarrow Pr(\varphi \text{ satisfied by random assignment}) > 0$

union bound

$\Rightarrow \exists$ a satisfying assignment for φ.

if $m < 2^k$.

<u>Also</u>: cannot do ~~much~~ any better.

if $m = 2^k$ could have all 2^k clauses on some k variables.

one must be false in <u>every</u> assignment

$\Rightarrow$ could be unsatisfiable.

Suppose we know more facts:

- suppose φ is k-CNF formula (m clauses, n vars)

  and each variable occurs on only L clauses. for some "small" L

  ↳ "locality" constraint.

Then can we say that φ is satisfied with many more clauses?

Thm: Sps. each var appears in $< 2^{k-2}/k$ clauses.

then φ is satisfiable indep of values of m, n.

Just depends on the "locality" value.

Follows from an important tool: the $\boxed{\text{Lovász Local Lemma}}$

(in a paper of Erdős and Lovász).

Needs some buildup.

Consider some probability space.

- Define a collection of good events

  $\mathcal{G} = (G_1, G_2 \ldots G_m)$

- for any subset $S \subseteq [m]$, let $\bigcap_{i \in S} G_i$ be written as G_S

- Define a dependency graph on these events $H = ([m], E)$

  such that $\forall i$, event G_i is independent of all events not in its neighborhood.

formally: for all S, T disjoint subsets of $[m] \setminus (\underbrace{\partial_H(i) \cup i}_{\text{i and its neighborhood}})$

$$\Pr(G_i) = \Pr\Big(G_i \,\Big|\, \big(\underbrace{\textstyle\bigcap_{j \in S} G_j}_{\text{all events in S happened}}\big) \cap \big(\underbrace{\textstyle\bigcap_{j \in T} \overline{G_j}}_{\text{all events in T did not happen}}\big)\Big)$$

Example: bad event B_i = clause i not satisfied by random x.

good event $G_i = \overline{B_i}$. = clause i satisfied by x.

then can define $E = \{(i,j) \mid$ clause i and clause j do not share variables $\}$.

Now: $\Pr$(clause i sat) $= \Pr$(clause i sat $\mid$ any settings of vars not in i)

$= \Pr(G_i \mid$ any events for clauses not in $i \cup \partial_H(i)\}$

Note: there dependency graph was simple to check.

Should be careful that we check that G_i is independent of all subsets S, T etc.

OK. Now the LLL:

Lovász Local Lemma (1975) Symmetric Version.

Let $G_1, G_2 \ldots G_m$ be events. Let $H = ([m], E)$ be dependency graph for them.

Pr(bad event) ≤ small

Suppose $Pr(\bar{G_i}) \le p \;\; \forall i$ And suppose max degree of $H \le d$.

degree of dependency small

$\Rightarrow Pr(\cap G_i) > 0$ as long as $pd \le 1/4$.

Pr(all good events happen) = 1 − Pr(some bad event happens)

much better than $p \cdot m < 1$ by union bound!!

let's prove Thm 1 (using the LLL)

$G_i = \{$clause i sat'd by random $x\}$. $\Rightarrow Pr(\bar{G_i}) = 1/2^k$.

~~also~~ $H = \{(i,j) \mid G_i$ and G_j share vars$\}$.

Now: G_i contains k vars

Each belongs to $\le 2^{k-2}/k$ clauses

$\Rightarrow G_i$ adjacent to $\le 2^{k-2}$ other G_j in H.

$\Rightarrow d \le 2^{k-2}$.

$\Rightarrow p \cdot d \le 2^{-k} \cdot 2^{k-2} = 1/4$ as desired

$\Rightarrow Pr(\cap G_i) = Pr($all clauses sat'd$) > 0$.

$\Rightarrow \exists$ a satisfying assignment ☺.

probabilistic method!

Great: Next steps.

1. Prove LLL.
2. Prove it again (via "witness" tree) — get an algo from it.
3. Hint of yet another proof (maybe). — using entropy.
4. Application to the Beck-Fiala Problem.

— x —

1. Original Proof of LLL was "not algorithmic".

Pf: define $G_S = \bigcap_{i \in S} G_i$.

Claim: Sps $S \subseteq [m]$ s.t. $Pr(G_S) > 0$. for $i \notin S$, $Pr(\overline{G_i} \mid G_S) \leq 2p$.

Pf: Induction on $|S|$. $|S| = 0$ is trivial, since $G_\emptyset = \Omega$ and $Pr(\overline{G_i}) \leq p$ by the assumption.

for $|S| \neq \emptyset$, let $T \subseteq S$ be nbrs of G_i, $U \subseteq S$ be non neighbors of G_i

$$Pr(\overline{G_i} \mid G_T \cap G_U) = \frac{Pr(\overline{G_i} \cap G_T \cap G_U)}{Pr(G_T \cap G_U)} = \frac{Pr(\overline{G_i} \cap G_T \mid G_U)\, \cancel{Pr(G_U)}}{Pr(G_T \mid G_U)\, \cancel{Pr(G_U)}}$$

$Pr(G_S) \neq 0 \Rightarrow Pr(G_U) \neq 0.$

$$\leq \frac{Pr(\overline{G_i} \mid G_U)}{Pr(G_T \mid G_U)} = \frac{Pr(\overline{G_i})}{Pr(G_T \mid G_U)}$$

smaller event

$\leq p$

union bound

$$1 - \sum_{j \in T} Pr(\overline{G_j} \mid G_U) \geq 1 - |T| \cdot 2p \geq 1/2.$$

$\leq d$

if $\exists j \in T \Rightarrow |U| < |S| \Rightarrow$ induction!!

$$\Rightarrow Pr(\overline{G_i} \mid G_S) \leq \frac{p}{1/2} = 2p.$$

Now: $Pr(G_{[m]}) = \prod_{i=1}^{m} Pr(G_i \mid G_{(j<i)}) \geq \prod_{i=1}^{m}(1-2p) > 0.$

$pd \leq 1/4 \Rightarrow p \leq 1/4$

———— x ————

OK:

Can improve LLL condition to say:

if $p(d+1) \leq 1/e \Rightarrow Pr(\bigcap_i G_i) > 0.$

similar proof see books

And this is best possible.

———— x ————

OK. Great. Show these amazing existential results (like for k-SAT)
and Packet routing
and Beck Fiala....

E.g:

Routing: Suppose graph. G.

s_i, t_i source-sink pairs $\quad P_i$: fixed path from source s_i to sink t_i

Want to send 1 packet each from s_i to t_i

Only one packet per edge per time.

Like hypercube routing, now fixed paths.

How long?

Must take: Dilation $D = \max_i |P_i|$ max path length.

Congestion $C := \max_e |\{i \mid P_i \text{ uses edge } e\}|$

Thm: [Leighton Maggs Rao]

see HW!

$\exists$ a routing algorithm that sends packets, and completes

in $O(C+D)$ steps !! (For any graph!)
(Any paths!)

So far: existential results, $\exists$ a good outcome ~~as~~ b/c $\Pr(\bigcap_{i\in[m]} G_i) > 0$.

But only showed $\Pr(\bigcap G_i) \geq (1-2p)^m$. Could be tiny!

Not algorithm that's efficient.

Let's fix that problem... (quite spectacularly)

——— X ———

Focus on k-SAT, same idea works in General.

A Trivial (?) Algorithm

(i) Start with a uniformly random truth assignment x.

(ii) while $\exists$ a clause C unsatisfied:

re flip all vars in C.

Does this even stop? (Even if each var in $\leq 2^{k-2}/k$ clauses).

$\boxed{\text{Thm 2: Trivial Algo stops in expected } O(m) \text{ time!!!}}$ [Moser – Tardos 2010]

Before we prove thm...

Same idea for many LLL applications. [see Moser-Tardos, other people's lecture notes... on webpage.]

[Sps each event G_i depends on some set $vars(i)$ of underlying rvs.

And $E = \{\{(i,j)\} \mid vars(i) \cap vars(j) \neq \emptyset\}$ be depending graph

Then trivial algo says:–

[while $\exists\, i$ st. G_i not satisfied:

re randomize vars in $vars(i)$.]]

Pf (for Thm 2)

Proceeds by building "witness" trees. Suppose $C_1, C_2, C_3, \dots, C_t, \dots$

be clauses reflipped by algo. (a clause may appear many times).

$\forall t$: build a witness tree T_t as follows.

- C_t is root
- for $i = t-1, t-2, \dots, 1$.

  if clause C_i has no vars in common with clauses in current tree, discard.

  Else add as child of deepest node it shares vars with.

Claim 1: Pr (some tree T_i appears) $\leq (2^{-k})^{|T_i|}$.

Pf: peel off some clause that is a leaf. It was reflipped.

$\Rightarrow$ not sat'd by orig assignment (with prob 2^{-k}).

remove it, look at next leaf (in peeled tree).

Either leaf in orig tree (so unsat in orig assignment, wp 2^{-k}).

or was unsat after its children were flipped.

But each var is again conditionally indep and unbiased

$\Rightarrow$ unsat wp 2^{-k}.

This happens for each of $|T|$ clauses $\Rightarrow Pr(\text{size } s \text{ tree}) \leq (2^{-k})^s$.

∎

Claim 2: # of possible trees of size $s \leq m \cdot \binom{(d+1)s}{s-1} \leq m(e(d+1))^s$

Pf: Suppose clauses are called $K_1, K_2, \ldots, K_m$ (in this order).
"klauses"

How to specify tree :-

① write down root (m possible options)

② Now traverse this tree in some order (say preorder)

for each node, write its children, smartly.

Note: by the way we built tree, each ~~tree~~ node (clauses)
children are some subset of $\underbrace{\text{its neighbors (or itself)}}_{(d+1)\text{ choices}}$
(no repeats).

So write down a $(d+1)$-bit vector indicating which are children.

$000..0 \Rightarrow$ no children

(apart from root)
$\Rightarrow$ tree is given by bit string of length $(s(d+1))$
with $(s-1)$ 1's in it.

total of $s-1$ children (since 1 root)

$\Rightarrow$ # trees $\leq m \cdot \binom{s(d+1)}{s-1} \leq m(e(d+1))^s$ trees.

$\Rightarrow$ Proof of LLL: each tree has prob $\leq (2^{-k})^s = p^s$.

and at most $m(e(d+1))^s$ trees.

$\Rightarrow E(\text{\# of trees seen by algo}) \leq \sum_s p^s \cdot m(e(d+1))^s$

$= m \cdot \sum_s (e \cdot p \cdot (d+1))^s = O(m)$

if $ep(d+1) < 1$.

☺

This kind of witness tree argument
also used for 2-choice proofs, etc.
Powerful technique.

The Beck-Fiala Conjecture & Discrepancy

Suppose subsets $S_1, S_2 \cdots S_m \subseteq [n]$.

coloring

We want to partition the elements of $[n]$ into two groups Red/Blue

such that $\underbrace{\max_i \left| (\text{Red} \cap S_i) - (\text{Blue} \cap S_i) \right|}_{\text{discrepancy of coloring}}$ is minimized

Why? { Pick a subset of half the people from the room s.t.
#people with each characteristic (gender / education / income)
are fairly preserved.

Can repeat (or use similar ideas) to try and get a representative subset of size S.... (see work on "Sortition")

— x —

OK:

- In Exercises from HW2,
  show that if random coloring, then
  discrepancy $\leq \sqrt{n \log m}$ whp.
  #elements #subsets

- Suppose we know that each person belongs to $\leq K$ subsets
  $\Rightarrow$ Beck Fiala Conjecture says:-
  discrepancy $\leq O(\sqrt{K})$

In fact Bansal-Jiang solve the case $K \geq \log^2 n$ recently!

$\boxed{\text{Open!}}$ $\boxed{\text{Best results show: } O(K) \text{ or } O(\sqrt{K \log \log n})}$

[BeckFiala] Bansal Jiang ~~FOCS~~ 25 FOCS

However, suppose

(a) Each person belongs to $\leq K$ subsets

(b) Each subset contains $\leq K$ people ← Extra condition

$\Rightarrow$ can show discrepancy $\leq O(\sqrt{k \log k})$ using the LLL.

Pf: (via LLL).

① Do a random coloring with each person being $\overset{+1}{R} / \overset{-1}{B}$ wp $\frac{1}{2}$ indep.

② $\Pr(\text{Set } S_i \text{ has discrepancy} \geq c\sqrt{k \log k})$

$$= \Pr\Big(|\underset{\substack{\uparrow \\ \text{Rademachers}}}{\textstyle\sum_j X_{ij}}| \geq c \cdot \sqrt{k \log k}\Big) \leq \exp\left(-\frac{c^2 (k \log k)}{2 \cdot K}\right) \doteq \frac{1}{K^{10}}$$

say if $c \geq \sqrt{20}$.

③ Define G_i = event that S_i has $\underbrace{\text{discrepancy} \leq c \cdot \sqrt{k \log k}}_{\text{"balanced"}}$

$\Rightarrow \Pr(\overline{G_i}) \leq 1/K^{10} = p$

Moreover define dependency graph

define $E = \{(i,j) \mid S_i \cap S_j \neq \emptyset\}$

then degree of this graph vertex i $\leq |S_i| \cdot$ #sets containing any element

$\leq K \cdot K = K^2$.

$\Rightarrow d \leq K^2$

$\Rightarrow p \cdot d \leq \frac{1}{4}$ for $k \geq 2$ $\Rightarrow$ LLL says $\exists$ solution where all sets "balanced".

Glimpse of Entropy-Compression Proof

Robin Moser's STOC 2010 talk
+ Terry Tao + Mary Wootters
blog notes.

Focus on k-SAT:

Important: if clause violated, every literal in it evals to False.
So by telling you the name of violated clause, I give you k bits of info.

But naively, need $\log m$ bits to say name.

let's name things smartly, using low degree.

— x —

- Let's run algo, and suppose it runs forever. (say $\geq T$ steps)
  $\Rightarrow$ use $\boxed{n + kT \text{ random bits}}$.

- Let's try to write random bits another way :-

TrivAlgo:
Pick n bits randomly for assignment
Let $K_1, K_2 \ldots K_j$ be violated
for these violated clauses $i = 1..j$
Print "fixing clause K_i"
Fix(K_i)
output final assignment.

Fix(K_i)
rerandomize bits in K_i
Suppose its children are
$K_i = K_0', K_1', K_2' \ldots K_d'$.
for violated children K_j'
print "jth child"
Fix (K_j')
output "wrapping up"

if we perform T fixes :- output

- $O(m \log m)$ bits in "fixing clause XXXX"
- $O(\log d)$ bits in "XX^{th} child" or "wrapping up" $\times T$ steps
- n bits at end.

observe: can recover the $Tk + n$ bits!

$\Rightarrow \boxed{m \log m + n + T \log d. \text{ bits.}}$

But cannot "compress randomness" — needs formalisation!!

Great:

So suppose we run for T steps. then. → we can time out after some T_0 steps.

We've taken $n + TK$ bits of randomness

And output $m \lg m + n + T(\lg d + O(1))$ bits of transcript

$\boxed{\text{And: its possible to infer the } n+TK \text{ bits from this transcript}}$!

Since $n+TK$ bits were truly random,

"cannot compress them".

(formally, any compression scheme for L bits must use $\geq L-c$ bits for all but 2^{-c} fraction of them)

So morally: (and we can formalize this, see notes on webpage): –

$$n+TK \leq m \lg m + n + T(\log d + \underset{\text{constant}}{c}) \text{ bits}$$

$\Rightarrow$ if $d = 2^{k-c-1}$ then

$$n + TK \leq m \lg m + n + T(k-1)$$

$$\Rightarrow T \leq m \lg m \text{ steps.}$$

Proof #2 (the Algorithmic proof) was one other way to capture this idea.

See notes for slightly different viewpoints. 2 approaches.

To wrap up:

LLL: collection of "bad events" $\{B_i\}_{i=1}^{m}$

(and associated good events $G_i = \overline{B_i}$)

Sps $Pr(B_i) \le p$

and $\deg_H(i) \le d \quad \forall i$

then if $pd \le 1/4$ (or $p(d+1) \le 1/e$)

$\Rightarrow Pr(\bigcap_{i=1}^{m} G_i) > 0$.

dependency graph $H = ([m], E)$

st B_i is indep of $[m] \setminus (N(i) \cup \{i\})$

- Better than union bound when applicable
- Elegant entropic / witness tree proofs to show algorithmic versions
- Used in
  - Packet Routing (see upcoming HW)
  - Beck Fiala bound
  - k-sat Algorithms
  - and other surprising applications....