2nd Moment Method

Today

- Lovasz Local Lemma

– no class Monday
– project preproposal due Monday

- Chebychev's Inequality
  $\forall \lambda > 0 \quad Pr(|X-\mu| \geq \lambda\sigma) \leq \frac{1}{\lambda^2}$

- Another version $\quad Pr(X=0) \leq \frac{Var(X)}{E(X)^2}$

Pf $\quad Pr(X=0) \leq Pr(|X-\mu| \geq \mu) \leq \frac{\sigma^2}{\mu^2} = \frac{Var(X)}{E(X)^2}$

Corollary: If $Var(X) = o(E(X))^2$ then $Pr(X>0) = 1-o(1)$

Another 2nd moment inequality

$$\boxed{Pr(X>0) \geq \frac{(E(X))^2}{E(X^2)}}$$

Follows from Cauchy-Schwartz

$$\boxed{[E(XY)]^2 \leq E(X^2)\,E(Y^2)}$$

Proof: wlog $E(X^2)>0$, $E(Y^2)>0$

Let $U = \frac{X}{\sqrt{E(X^2)}}$ $\quad V = \frac{Y}{\sqrt{E(Y^2)}}$

$2|UV| \leq U^2 + V^2$

$\Rightarrow 2|E(UV)| \leq 2E(|UV|)$

$\leq E(U^2) + E(V^2) = 2$

$\Rightarrow [E(UV)]^2 \leq 1$

$\equiv [E(XY)]^2 \leq E(X^2)E(Y^2)$

Set $Y = 1_{X>0}$

$$[E(X)]^2 \leq E(X^2)\,\underbrace{E[(1_{X\geq 0})^2]}_{Pr(X>0)}$$

Lovász Local Lemma

Let $E_1, E_2, \ldots, E_n$ be set of "bad" events $\qquad$ $\Pr(E_i) < 1 \quad \forall i$ $\quad$ p.

Say want to show $\Pr\left(\bigcap_{i=1}^{m} \overline{E_i}\right) > 0$ $\qquad$ "positive probability that nothing bad happens"

2 cases where easy:

1. E_i are mutually independent $\qquad (1-p)^n$
2. $\sum_{i=1}^{n} \Pr(E_i) < 1$ $\qquad$ union bound suffices

LLL is clever comb

Defn E mutually indep of $E_1, \ldots, E_n$ if $\forall$ subset $I \subseteq [1..n]$

$$\Pr\left(E \mid \bigcap_{j \in I} E_j\right) = \Pr(E)$$

Defn A dependency graph for $E_1, E_2, \ldots, E_n$ is $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ & E_i is mutually indep of $\{E_j \mid (i,j) \notin E\}$

Lovász Local Lemma

Let $E_1, \ldots, E_n$ be set of events s.t.

1. $\Pr(E_i) < p \quad \forall i$ ✓
2. The max degree in dependency graph is d ✓
3. $4dp \leq 1$ ✓

Then $\Pr\left(\bigcap_{i=1}^{n} \overline{E_i}\right) > 0$

several variants & generalizations (see notes)

$\boxed{\text{Application: k-SAT}}$

Let φ be a $\underbrace{\text{k-SAT formula}}_{\text{each clause has k literals}}$ w/ n vars
m clauses.

$x_1 \vee \bar{x}_3 \vee x_5$

$\Pr(\text{random assignment to vars satisfies a particular clause}) = 1 - \frac{1}{2^k}$

$\Pr(\exists \text{ unsatisfied clause}) \leq m \frac{1}{2^k}$

If $m < 2^k \implies \exists$ satisfying assignment.

$\boxed{\text{Thm:}}$ Let φ be a k-SAT formula w/ n vars
m clauses.
If no var appears in $> T = \frac{2^k}{4k}$ clauses,
then formula has satisfying assignment

Pf LLL

E_i event that clause i not satisfied

$p = \Pr(E_i) = 2^{-k}$

E_i is mutually indep of any clause it doesn't share vars with

$d \leq kT = k \cdot \frac{2^k}{4k} = \frac{2^k}{4}$

$4dp = 4 \cdot \frac{2^k}{4} 2^{-k} \leq 1$ ✓

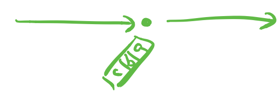
Application 2: Packet Routing

graph; n packets
each packet has
s_i source
t_i destination

and specific path P_i $\quad s_i \xrightarrow{P_i} t_i$

only one packet can traverse an edge per time unit

so



Schedule specifies for each packet when to move, when to wait

$d = \max_i |P_i|$ dilation

$c = \max_e (\#\text{paths } P_i \text{ that use } e)$ congestion

How long for each packet to reach its destination?

$\Omega(c+d)$??? $O(cd)$

[Leighton, Rao, Maggs] $\exists$ schedule of length $O(c+d)$ always
indep of n!

Can be proved using LLL

High level idea:

for each packet, assign random initial delay
in $[1, \alpha(c+d)]$

guarantees limited dependency between congestion
on different edges in different time periods

Algorithmic version

[Moser, Tardos]

2 clauses C_i & C_j are <u>dependent</u> if they share a var

$D(C_i) \triangleq \{C_j \mid C_i \text{ \& } C_j \text{ dependent}\}$ Let $d = \max_i |D(C_i)|$

Thm Let φ be a k-SAT formula with $d \le \frac{2^k}{8}$ (m clauses, n vars)
Then φ is satisfiable & a satisfying assignment can be found in poly time.

Super cool proof $C_1, C_2, \ldots, C_m$

Algorithm

Initialize $\vec{x} = (x_1, \ldots, x_n)$
where $x_i = \begin{cases} T & \text{w.p. } \frac{1}{2} \\ F & \text{w.p. } \frac{1}{2} \end{cases}$

While $\exists$ clause C that is not satisfied
  Fix(C)

Fix(C)

Randomly reassign k vars in C to T/F (indep w.prob $\frac{1}{2}$)
$\Rightarrow$ gives updated $\vec{x}$

While some clause D of φ that shares vars w/ C is violated
  Fix(D)

[note D could be C]

Always process clauses in fixed order

Alg

$\text{Fix}(C_1)$ $\text{Fix}(C_2)$ . . . $\text{Fix}(C_m)$ $\le m$ level 1 calls

Observations

1. # random bits used is $n + k \cdot \#$ calls to Fix

2. If Fix(C) terminates, then it terminates with an assignment in which all clauses in D(C) are satisfied.

3. $S = \{C_j \mid \text{satisfied before a top level call to Fix } C_i\}$

if Fix(C_i) terminates, then all clauses in S are still satisfied.

$\Rightarrow$ make progress when outer calls finish.

Thm $\forall$ kSAT formula w/ $d \le \frac{2^k}{8}$ alg terminates in polytime w.h.p.

Pf $f: A \to B$ injective $|B| \ge |A|$

Suppose abort the computation after T calls to Fix (if not done.)

ALG uses up to $n+kT$ bits.

Let $A = \{0,1\}^{n+kT}$

$|A| = 2^{n+kT}$

write down transcript of computation for fixed x_0, y_0

$$x_0, y_0, \varepsilon \xrightarrow{Fix(C_1)} \underbrace{x_1, y_1, z_1} \xrightarrow{Fix(C_2)} x_2, y_2, z_2 \longrightarrow \cdots \longrightarrow x_T, \varepsilon, z_T$$

final bits — description of comp

- # bits used in $x_{t+1}, y_{t+1}, z_{t+1}$ $<$ # bits used in x_t, y_t, z_t
- process is reversible

$$f(x_0, y_0, \varepsilon) \longrightarrow (x_T, \varepsilon, z_T)$$

$$\overset{t \text{ steps}}{(x, y, z)} \xrightarrow{Fix(C_i)} (x', y', z')$$

z' is obtained from z by appending

- 1 binary rep of C_i if outer call to C_i $\lceil \log_2(m) \rceil + 1$
- 1 "binary rep of C_i" if inner call to C_i (from C_j) $\lceil \log_2(d) \rceil + 1$ bits.

C_j intersects $\underbrace{C_{i_1}\ C_{i_2} \ldots C_{i_d}}$

- add a 0 if all clauses in $D(C_i)$ are satisfied.

Claim: transcript is reversible

$x_1 \vee \bar{x}_2 \vee x_5$

① construct

$$x, y, z \xrightarrow{Fix(C)} (x', y', z')$$

$x_1 = 0$
$x_2 = 1$
$x_5 = 0$

newly set values of C

$$f(x_0, y_0, \varepsilon) \longrightarrow (x_T, \varepsilon, z_T)$$

$$n + kT \text{ bits} \implies n + \underbrace{\text{bits for outer calls}}_{m(\lceil \log_2(m) \rceil + 2)} + \underbrace{\text{bits for inner calls}}$$

$$T(\lceil \log_2(d) \rceil + 2) \qquad \leq 0$$

$$\leq k - 3 \qquad d = \frac{2^k}{8}$$

$$n + kT \implies \#\text{bits in final transcript} \leq n + m(\lceil \log_2(m) \rceil + 2) + T(k-1)$$

$\forall$ input ALG doesn't terminate

$$\#\text{inputs} = \#\text{outputs}$$

$$2^{n+kT} \leq 2^{n + m(\lceil \log_2(m) \rceil + 2) + T(k-1)}$$

$$\implies T \leq \underbrace{m(\lceil \log_2(m) \rceil + 2)}_{c}$$

random

$\underbrace{\Rightarrow}_{\substack{n+kT \text{ bits} \\ \text{alg uses}}}$ If $T > S$ $\exists$ input on which ALG terminates

suppose that on fraction $\geq 2^{-c}$ of inputs ALG doesn't terminate

$$2^{n+Tk-c} \leq \boxed{\text{\# inputs on which it doesn't terminate}} \leq 2^{n+m(\lceil \log_2(m) \rceil + 2) + T(k-1)}$$

$$\Rightarrow \quad T \leq \underbrace{m(\lceil \log_2(m) \rceil + 2) + c}_{S'}$$

If $T > S'$ then alg doesn't terminate on $< 2^{-c}$ fraction of inputs

succeed w.p. $\geq 1 - 2^{-c}$ ∎