Back to k-SAT

Easier cases?

Thm Any k-CNF formula s.t. each var occurs $\lesssim \frac{2^k}{ek}$ times is satisfiable.

Pf Idea - probabilistic method i.e. take rand. assignment!

but how to analyze?

Lovasz Local Lemma ('77)

Let $E_1, \dots, E_m$ be events s.t. each E_i depends on $\leq d$ other events.

Suppose $\Pr(E_i^c) \lesssim \frac{1}{ed}$. $= p$.

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

[if completely indep, trivial)
(in general, if $\Pr(E_i^c) < \frac{1}{m}$, can apply union bd]

Apply LLL with $E_i = [i^{th}$ clause is satisfied$]$

$$d \lesssim k \cdot \frac{2^k}{ek} = \frac{2^k}{e}.$$

$$\Pr(E_i^c) = \frac{1}{2^k} \lesssim \frac{1}{ed}.$$

$$\Pr\left(\bigcap_{i=1}^m E_i\right) > 0.$$

Pf: **Claim** $\forall$ subset $S \subseteq \{1 \dots m\}$

$$\Pr\left(E_i^c \,\middle|\, \bigcap_{j \in S} E_j\right) \leq \alpha$$

(for some fixed α with $p \leq \alpha < 1$.)

LLL follows since

$$\Pr\left(\bigcap_{i=1}^m E_i\right) = \Pr(E_1)\Pr(E_2|E_1)\Pr(E_3|E_1 \cap E_2) \cdots$$

$$\geq (1-\alpha)(1-\alpha)(1-\alpha) \cdots$$

$$= (1-\alpha)^m > 0. \longrightarrow \left(1 - \frac{1}{d}\right)^m$$

$\left(\Pr(X|Y) = \frac{\Pr(X \cap Y)}{\Pr(Y)}\right)$

Pf of claim: By induction on $|S|$

$$= (1-\alpha)^{\cdots} > 0. \quad \to \left(1-\frac{1}{d}\right)$$

Pf of Claim: By induction on $|S|$.

$S = \phi$: $\Pr(E_i^c) \le p \le \alpha$. ✓

Fix i. Let $A = \{ j \in S : E_j \text{ depends on } E_i \}$ $(|A| \le d)$.

$B = S - A$.

$$\Pr\left(E_i^c \,\middle|\, \bigcap_{j\in S} E_j\right) = \frac{\Pr\left(E_i^c \cap \bigcap_{j\in A} E_j \,\middle|\, \bigcap_{j\in B} E_j\right)}{\Pr\left(\bigcap_{j\in A} E_j \,\middle|\, \bigcap_{j\in B} E_j\right)} \le \frac{p}{(1-\alpha)^d}$$

$$\text{numerator} \le \Pr\left(E_i^c \,\middle|\, \bigcap_{j\in B} E_j\right)$$

$$= \Pr(E_i^c) \le p.$$

$$\text{denominator} = \Pr\left(E_{i_1} \,\middle|\, \bigcap_{j\in B} E_j\right) \cdot \Pr\left(E_{i_2} \,\middle|\, E_{i_1} \cap \bigcap_{j\in B} E_j\right) \cdot \Pr\left(E_{i_3} \,\middle|\, E_{i_1} \cap E_{i_2} \cap \bigcap_{j\in B} E_j\right) \cdots$$

write $A = \{i_1 \dots i_k\}$, $k \le d$

$$\ge (1-\alpha)(1-\alpha)\cdots \quad \text{by ind. hyp.}$$

$$= (1-\alpha)^k$$

$$\ge (1-\alpha)^d$$

$$\Rightarrow \Pr\left(E_i^c \,\middle|\, \bigcap_{j\in S} E_j\right) \le \frac{p}{(1-\alpha)^d} = \alpha$$

by choosing $p = \alpha(1-\alpha)^d$.

Set $\alpha \approx \frac{1}{d}$ and $p = \frac{1}{d}\left(1-\frac{1}{d}\right)^d \to \frac{1}{ed}$.

$\square$

(General "asymmetric" form of LLL:

Suppose $\exists \alpha_i \in (0,1)$ s.t.

(General asymmetric [illegible]

Suppose $\exists \alpha_i \in (0,1)$ s.t.

$$\Pr(E_i^c) \le \alpha_i \prod_{j:\, E_j \text{ depends on } E_i} (1-\alpha_j)$$

Then $\Pr\left(\bigcap_{i=1}^{m} E_i\right) \ge \prod_{i=1}^{m} (1-\alpha_i)$.)

But above pf does not yield efficient alg'm even with rand.

(need $\left(\frac{1}{1-1/d}\right)^m$ trials! $\rightarrow$ worse than brute force!)

Moser's Rand. Alg'm ('08)

Given k-CNF formula F s.t. each var occurs $\lesssim \frac{2^k}{c_0 k}$ times

Solve(F):

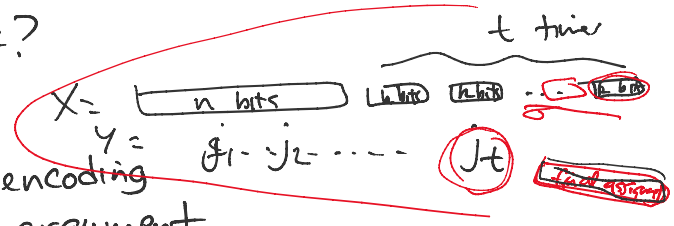
1. pick rand. assignment
2. while $\exists$ unsatisfied clause C
3. Fix(C) $\leftarrow \le m$ choices

Fix(C): // make C satisfied but keep all prev'ly satisfied clauses satisfied

1. replace all k vars in C with new rand. values
2. while $\exists$ clause C' that shares var with C & is not satisfied $\leftarrow$ # choices for C' $\lesssim \frac{2^k}{c_0} = d$.
3. Fix(C')

if it terminates, it finds sat. assignment

But will it terminate?



Analysis:

idea - counting or encoding argument

Suppose $\Pr[\text{alg'm needs} \geq t \text{ calls to Fix}] > 1/2$.

Run alg'm & stop after t calls to Fix.

Define X = sequence of rand. bits used by alg'm

$\Rightarrow$ # bits is $\underbrace{n}_{\text{init. rand assignment}} + \underbrace{kt}_{k \text{ rand bits per call to Fix}}$

Define Y = sequence of the clauses fixed together with final assignment

$\Rightarrow$ # bits is $\leq \underbrace{m(\log m + O(1))}_{\text{from Solve, } \leq m \text{ choices}} + \underbrace{t(\log d}_{\text{from Fix, } \leq d \text{ choices}} + \underbrace{O(1)}_{\text{extra bit to indicate end of recursive call}}) + n$

Obs From Y, can uniquely recover X.

Pf: just run alg'm backwards! $\square$

$\Rightarrow$ # possible X's $\leq$ # possible Y's

$\Rightarrow \frac{1}{2} \cdot 2^{n+kt} \leq 2^{m(\log m + O(1)) + t(\log d + O(1)) + n}$

$\Rightarrow \cancel{n} + kt \leq m(\log m + O(1)) + t(\log d + O(1)) + \cancel{n}$

$(k - \log d - O(1))t \leq O(m \log m)$

Set $d = \frac{2^k}{c_0}$

$\log d = k - \log c_0$

$$\underbrace{(k - \log d - O(1))}_{\text{positive}} t \leq O(m \log m)$$

$$\Rightarrow \quad t \leq O(m \log m)$$

$\Rightarrow$ algm find sat assignment in polytime w. prob $\geq \frac{1}{2}$.

$\square$