22C: 296 - Seminar on Randomization
Lecture 10 (24-09-03)
Scribe: Saurav Pandit

Recap:

We showed using Lovasz Local Lemma that any instance of k-SAT($2^{\frac{k}{50}}$) is satisfiable.

Now, we want to design an "efficient" algorithm to find a satisfying truth-assignment for any given instance of k-SAT($2^{\frac{k}{50}}$).

Beck(1991) designed an algorithm for this problem that is poly-time in m (no. of clauses) but exponential in k. Assuming that k=O(1), we got a poly-time algorithm for the problem.

ALGORITHM:

Stage 1

Consider the variables in some order & to each variable not belonging to a *dangerous* clause, assign T or F randomly with equal probability.

[A clause C is "dangerous" if

(i) C has exactly k/2 assigned literals.

(ii) C is not satisfied.]

If a variable belongs to a dangerous clause, skip it.

Lemma:

Let G be the graph whose vertex set is the set of all clauses and whose edge-set is $\{\{C_i , C_j\} \mid C_i$ & C_j share a variable$\}$. Let H be the subgraph of G whose vertices are clauses surviving stage 1 and whose edge-set is :

$\{\{C_i , C_j\} \mid C_i , C_j \in V(H)$, C_i & C_j share a variable unassigned in Stage 1$\}$

Then with probabilty 1 - o(1) the size of any connected component of H is $\leq$ z . log(n) for some constant z.

Proof:

Let C_1 , C_2, ... , C_r be a set of clauses such that for any i , j , $dist_G(C_i , C_j) \geq 4$. For C_i to survive there must be a dangerous clause $D_i \in \{C_i\} \bigcup N(C_i)$.

To each C_i assign a dangerous clause $D_i \in \{C_i\} \bigcup N(C_i)$. Since $dist_G(C_i , D_i) \leq 1$, $dist_G(D_i , D_j) \geq 2$ for i $\neq$j. Hence D_i& D_j do not share any variables.

Prob[All D_i 's have become dangerous]

$= \prod$ Prob [D_i has become dangerous]
$= (\frac{1}{2^{\frac{k}{2}}})^r$
$= 2^{\frac{-kr}{2}}$.

Prob[C_i 's have all survived]
= Prob[$\exists$ $D_i \in \{C_i\} \bigcup N(C_i)$ that has become dangerous for each i]
$\leq \sum 2^{\frac{-kr}{2}}$ . . . (*by union-bound*)
$= (d+1)^r.\ 2^{\frac{-kr}{2}}$
$= ((d+1).2^{\frac{-k}{2}})^r$ . . . .(1)

DEFINITION:

A subset T of clauses is called a ***4-tree*** if
(i) $dist_G(C,C') \geq 4$ for any $C,C' \in T$, $C \neq C'$.
(ii) If we form a graph on T connecting pair of vertices at distance exactly 4, then T is connected.

Consider a graph G_4 whose vertex-set is all clauses & edge-set is $\{\{C_i , C_j\} \mid dist_G(C_i , C_j) = 4\}$.

Note that any 4-tree T induces a connected sub-graph in G_4. Therefore number of 4-trees is bounded above by number of sub-graphs of G_4.

QUESTION:

How many connected sub-graps does G_4 have?

ANSWER:

The degree of each vertex in G_4 is $O(d^4)$.

We can show that any n-vertex t-regular graph has at most $n.t^{2.r}$ connected sub-graphs of size n... (*Proof skipped for now*)

Using this result we get for some constant a, G_4 has $\leq a.m.d^{8.r}$ connected sub-graphs.

Hence, the number of 4-trees in g is $\leq a.m.d^{8.r}$ for some constant a. . . . .(2)

Therefore , Prob[Some 4-tree of size r will survive] $\leq (a.m.d^{8.r}).((d+1).2^{\frac{-k}{2}})^r$ . . . (*by union-bound*)

By substituting $r = b.\log(m)$ for some constant b, we can show that this quantity is o(1)... (*Calculations skipped for now*)

Hence, Prob[There exists a 4-tree in H of size $> b.\log(m)$] $\leq o(1)$.

From this we have to show that :

Prob[There exists a connected component in H of size $> z.\log(m)$] $\leq o(1)$.