Undergraduate Complexity Theory
Lecture 26: Beyond Worst-Case Analysis

Marcythm

July 29, 2022

1 Lecture Notes

Assume $\mathsf{P} \neq \mathsf{NP}$ for the rest lectures.
So $\mathsf{3SAT} \notin \mathsf{P}$. Possibilities:

1. Allow more than $\text{poly}(n)$ time to solve all instances.
2. Relax correctness: "Approx Algos", maybe $\exists$ poly-time A s.t. $\forall \phi \in \mathsf{3SAT}$, $A(\phi)$ outputs an assignment satisfying 90% of clauses.
3. Look for poly-time algo that correct on "most" inputs, may be $\geq 99\%$ of all 3CNFs.

Dream: Show hardness for 1, 2, 3, just assuming $\mathsf{P} \neq \mathsf{NP}$.
Backup Dream: Make one new basic assumption, try to derive many consequences.

For point 1:
E.T.H.: $\exists \delta_0 > 0 : \mathsf{3SAT} \notin \mathsf{TIME}(2^{\delta_0 n})$, active field of research right now, conseqs of this in next lec.
If ETH is true, then no algo solve LCS in $O(n^2)$ time.
$\tilde{\mathsf{E}}\mathsf{TH} \implies \mathsf{BPP} = \mathsf{P}$ (mentioned in lec 22).

For point 2: partially realized.
PCP Theorem ('93): $\mathsf{P} \neq \mathsf{NP} \implies$ not exists poly-time A s.t. $\forall \phi \in \mathsf{3SAT}$, $A(\phi)$ outputs assignment satisfying $\geq 99.999\ldots\%$ of clauses.
Håstad'99 get this fraction down to $7/8 + \epsilon$. (optimal hardness result, since actually exists an algo that always output assignment satisfying $7/8$ of clauses.)

Fact 1.1. *$\exists$ efficient randomize algo A s.t. $\mathbf{E}[\#$ clauses satisfied by $A(\phi)] \geq 7m/8$.*

Lemma 1.2. *Let $\phi(x)$ be a E3CNF, let $a \in \{0,1\}^n$ be a uniformly random assignment, then*

$$\mathbf{E}[\# \textit{ clauses sat'd by } a] = \frac{7}{8}m$$

Proof. Let $I_j = \begin{cases} 1, & \text{if } a \text{ sats the } j\text{th clause of } \phi \\ 0, & \text{otherwise} \end{cases}$, $j = 1\ldots,m$. Then

$$\mathbf{E}[\#] = \mathbf{E}[I_1] + \mathbf{E}[I_2] + \cdots + \mathbf{E}[I_n] = \sum_{i=1}^{m} \mathbf{Pr}[i\text{th clause is sat'd by } x = a]$$

For each clause, it's in the form $x_i \vee x_j \vee \overline{x_k}$, so the probability it's sat'd is $7/8$. Thus

$$\mathbf{E}[\#] = \frac{7}{8}m$$

$\square$

Min-Bisection: Given G, find $S \subseteq V, |S| = n/2$ s.t. # edges between S and $\overline{S}$ is minimized. NP-hard in '70s, and in 2017: no known poly-time algo that achieves $\leq C \times \min$ (but can achieve $\leq \log^2 n \times \min$). Assume $\mathsf{P} \neq \mathsf{NP}$, we can't rule outpoly-time algo getting factor $1 + \epsilon$.

Remark 1.3. The reductions between (NP-complete) problems only preserve exact satisfiability, but not preserve approximation quality.

Worst-Case Hardness: $\forall$ poly-time SAT solvers, $\exists$ poly-time formula generator s.t. `Solve(Gen(n))` fails (for sufficiently large n).

Average-Case Hardness: $\exists$ poly-time formula generator, $\forall$ poly-time SAT solvers `Solve(Gen(n))` fails (with high probability) (for sufficiently large n). Here the generation algo always allowed to be randomized, since otherwise the solver can hard-code the solution.

Average-Case Hardness is always desirable, for cryptography.

Definition 1.4 (Uniformly Random E3SAT Instances)**.** $n = \#$ vars, $\Delta =$ "clause density", $m := \Delta n$.

$$\phi = \text{Choose } m \text{ random clauses independently, each clause is uniformly chosen from all } 2^3\binom{n}{3} \text{ possibilities}$$

Q0: Is ϕ likely to be sat or unsat? Depends on Δ.

Exercise 1.5. *If $\Delta > 5.2$, then ϕ is exponentially unlikely to be sat.*

Theorem 1.6. *If $\Delta \leq 0.16$, then ϕ is exponentially unlikely to be unsat.*

Conjecture 1.7. $\exists$ *"phase transition" at $\Delta \approx 4.2667$, i.e. a sharp decrease of probability.*

Exercise 1.8 (Chernoff bound)**.** *If $\Delta \geq 10$ish, w.h.p. ϕ is unsat, and every assignment $a \in \{0,1\}^n$ sats $\leq 7/8 + \epsilon$ fraction of constraints.*

Algo hardness: if $\Delta < 4.2667\cdots$: solver should find a sat assignment. 2017: poly-time algo that provably works when $\Delta \leq 3.52$. Algos that seem to work in poly-time for $\Delta \leq 4.266\cdots$.
If $\Delta > 4.2667\cdots$: solver should give a proof that ϕ is unsat.

Definition 1.9. A Δ-3SAT-Refuter is a poly-time algo that: given any ϕ, either outputs "unsat" or "no comment"; is never wrong. Given random ϕ with param Δ, $\mathbf{Pr}[A(\phi) = \text{unsat}] \geq 99\%$.

In 2017: do $\Delta-$-3SAT-Refuters exist for $\Delta = C$? Unknown. But for $\Delta = \sqrt{n}$, it exists.

Hypothesis 1.10 (Feige's Hypothesis)**.** $\forall$ *constant Δ, $\Delta -$ 3SAT-Refuter don't exist.*

This hypothesis directly implies that $\mathsf{P} \neq \mathsf{NP}$ (or there will exist poly-time algo for 3SAT, thus such refuter exists), Håstad's result (left as exercise), and no poly-time approx for Min-Bisection problem achieving $\leq 4/3$.