Lecture 24 - Hardness

We'll talk about a bunch of hard problems...

Well, we don't know how to prove anything is hard, so it's always...

Hardness: Assumption + Reductions

= algorithms

All hardness research is arguably just

research.

eg: 3Sat \notin P

\rightarrow \Phi \text{ Indep-set} \notin P

\text{NP} \notin \text{P}

\rightarrow \text{NP} \notin \text{P}

then: \text{4-Sat} \Rightarrow G \text{ has I.S. of size } \leq k

\text{4-unSAT} \Rightarrow G \text{ doesn't have } \ldots

Yes, there's a wonderful thing of NP-completeness based on
but insufficient.

"worst-case hardness": better if \exists algorithms for generating
hard instances

How hard? \text{E-Time}(2^{\sqrt[n]{n}}) = \text{E-Time}(2^{2^{\sqrt[n]{n}}}) \neq O(3^n)?

But remember, can't prove anything... need more assumptions. Try to be economical.

Last time: CP \rightarrow \text{LWE} \rightarrow \text{Cryptography}

Worst-case \rightarrow \text{poly-time} \rightarrow \text{quantum}

(Worst-case) \Rightarrow \text{approx} \Rightarrow \text{hard} \Rightarrow \text{poly}

Worst-case \approx \text{easy} \approx \text{easy} \approx \text{easy} \approx \text{easy}

solving nice diagonal
and \text{poly-time}
LPN: the mod 2 version. Parameters $n, \beta \in [0, 1/2) \Rightarrow 2^{\beta n}$ chosen uniformly.

- Alg can ask for (noisy) $F_2$-lin eqn:
  - Gets $a_1s_1 + \ldots + a_n s_n = b$, where $a_i \in F_2^n$ uniform, and $b$ is correct val. except with prob. $2^{\beta n}$.

Task: adapts.

Assumption: if $\beta$ is large, $2^{\beta n}$ is not solvable in poly-time (for worse).

$\Rightarrow$ OWFs, very efficient SKE ("minicrypt")

$\Rightarrow$ PKE $(\text{Ale'03}) \Rightarrow$ PKG if take $\beta = \Theta(n^{\frac{1}{2}})$ [Lys'05, [Wu'09, ? local sample]]

Fastest alg: $\{ \text{BKW'03} \} \text{ "Clever Gaussian Ell" } [\text{Quite beautiful}]

\begin{align*}
\text{Time: } & 2^{O(n^{1.2})} \quad \text{ [even for } \beta = 2^{-n^{0.9}} \text{]} \quad \text{samples} \\
\text{Rem: } & \text{Fastest alg using only poly(n) eq's: } 2^{O(n/\log(n))} \quad \text{[Lys'05, Wu'09]}
\end{align*}

\textit{Rem: } [Katz-Smith'11]: search = decision = dist. from random.

\begin{align*}
\text{[Fei'02, Ale'03]} \quad & \text{Sparse-LPN: } \text{Fix } k > 3. \text{ Eq's: } a_1s_1 + \ldots + a_n s_n = b \text{ have only } k \text{ nonzero } a_i.
\end{align*}

\textit{Rem: } Alg gets $m$ examples, poly-time $T$, all.

- If $m > nk$, can get $\Theta(nk^2)$ eq's, many times.

- Poly-time alg even for $m = \Theta(n^{k/2})$: [Fei'06, OW'07]

- e.g. Assump: for $k = 3$, no poly-time alg. for even $2^{o(n^{\log n})}$?

\begin{align*}
\text{m} = O(n) \quad \text{can even find } \exists \text{ with } \frac{1}{2} + \frac{1}{2} \beta \text{ agree. w.r.t. } \text{v}.
\end{align*}
The Lasserre/Parrilo/SOS hierarchy doesn't solve until day \( \mathcal{O}(n) \) (time \( 2^{\mathcal{O}(n)} \)).

**Claim** ([Håstad '99]) \( \frac{3}{4 + \varepsilon} \)-approx, 3LIN(odd) is \( \text{NP-hard} \).

\[ 3\text{SAT} \quad \text{implies} \quad (\mathcal{O}(n)) \quad \text{-approx Max-3Sat.} \quad \text{Parallel RES "Label-Cover"} \]

**Moshkovitz-Raz '10**

Given \( \mathcal{O}(n) \) q-ans over \( q \)-bits

\[ x_i + x_j + x_k = 0 \quad \text{mod} \quad 2 \]

NP-hard to tell if \( \geq \left(\frac{1}{4} - \varepsilon\right) \) unsat. or eq. \( \frac{3}{1 + \varepsilon} \)

\( \leq \left(\frac{1}{4} + \varepsilon\right) \)

**“easy-case”**

Sparse-LP assumption, gives natural, effic. way to generate hard-seeming insts.

[Stronger assume than \( \text{P\#P} \), of course, but yet more!]

Great basis for further reductions

\[ \text{GED known Max-Cut, QSat approx} \]

\[ [\text{Håstad, TSSW}] \quad \text{w/o the CSB} \]

\[ \text{TSP approx} \quad [\text{PL'06}] \]

Dense k-Subgraph, Ham-unif. Sparsest Cut,

GISo hardness...

Yields poly-time gen’le (arguably "unnatural") hard insts for...

Let's talk about 3SAT, much better studied, anote papers, heuristics algs for it.

Generating "hard" satisfiable 3SAT instances trickier.

[planted? \( \Rightarrow \) if \( s_i = 0 \), more likely to see \( s_i \).]
Q: Suppose choose a unit rand. m-clause, n- var 3sat inst.
Is it satble?

A: Empirically/stat phys: \[I \approx 4.2\] s.t.
\[
m > cn \Rightarrow \text{unsat w.h.p.} \quad \text{Thm: true for } c \approx 4.5
\]
\[
m < cn \Rightarrow \text{sat w.h.p.} \quad \text{Thm: true for } c \approx 3.5.
\]

Is Assump? [not very useful]: For \(c = 4.2\), and 3sat rep. \(2^{(m/n)}\) time.

Better: Feige's R3Sat assump: For all suff. large const. \(c\):
\[
\P \text{ a poly alg. which relates almost all} \\
\text{random 3sat insts w/ } m \approx (cn).
\]

[outputs a proof of unsat; alternatively, is weaker;

just have to say "typical" w.h.p., but need

known to be broken if \(c = 2^{(n^{1/3})}\)]

As w/3Lin, say "typical" if sat"ble".

CNF-3Sat worst-case \(\Omega(2^n)\): Trivial alg: \(2^n \text{poly}(m)\) time.

3-Sat: \(\tilde{O}(1.3^n)\) best known.

- eas\(y\) \(\tilde{O}(1.3^n)\) alg: Walk-Sat [Schöning'99]
  \[\begin{align*}
  \text{1. pick rand. assgn} \\
  \text{2. take any unsat clause, flip rand. bit in it} \\
  \text{Repeat (2) \(O(n)\) times to try to find sat. assgn} \\
  \text{Repeat (2) \(O(\sqrt{n})\) times.}
  \]
ETH: \[ f > 0 \text{ s.t. } \exists \text{ 3-Sat reg in time } 2^{f(n)} \text{ time} \]

**Imp:** Patrascu'08

Once you have such a hypothesis, reduction size/time is key.

**Jai:** e.g., you prove Subset-Sum is NP-hard. I

Assuming ETH, implies \( \exists f \) on size-\( N \) inputs reg in time \( 2^{f(N)} \) time.

**Planar-Ham-Path**

Input size \( (n^2) \) (nodes+edges)

Alg. effic. of reductions is important!

**Jai:** Some NP-complete jobs are solvable in \( 2^{n^{0.02}} \) time.

e.g.: \( \text{Given } G, \exists \text{ an indep set of size } \approx n^{0.05} \)?

**rem:** "most" classic NPC reductions have output size \( O(n) \)

e.g.: ETH \( \Rightarrow \) 3-Coloring, Ham-Path, Indep. Set reg in \( 2^{O(n)} \) time.

\[ \text{ETH} \Rightarrow \text{CNF-Sat not solvable in } 2^{(1-\varepsilon)n} \text{ poly}(m) \text{ time.} \]

**Facts:**
- \( \exists 1.3^k \text{ poly}(n) \) alg for "Does \( G \) have \( \{0,1\} \text{-} \text{VC of size } k?".
- But ETH \( \Rightarrow \) no \( 2^{o(n)} \text{ poly}(n) \) alg.
- \( \exists n^{O(k)} \text{ alg for } k\text{-Clique. ETH} \Rightarrow \) no \( f(k)n^{o(k)} \) alg.
- Max-Cut solvable in time \( 2^{O(\text{tw}(G))} \) poly(n).
- ETH \( \Rightarrow \) no \( 2^{1-o(1)n} \) poly(n).
(Patrascu-Williams) \( SETH \Rightarrow \text{"k-Sum" on n integers} \)

\( \exists \text{ some } k \text{ adding to } \Omega(n^k) \) \( \text{reg} \text{ s} \text{ in} \text{ time} \)

\( \text{for } k = \Omega(n^{\log n}) \text{ handling the fabled 3-Sum would be cool.} \)

Hardness of approx:

Mother of all probs: \( \text{Label Cover} (q) \)

\[ \text{LC: inapprox \ : \ 3Sat: NPC} \]

ACSP. Input:

\[ V \quad \text{\# each edge has a \# fcn} \]

\[ (\text{binary}) \quad \text{\# \# \# written on it} \]

\[ A(v) = \prod u \in \text{deg}(v) - \text{\# constr sat if } A(u) = II(u) \]

\[ \text{\# Projection constrs.} \]

\[ \text{Thm [Raz'94]} \quad \forall \varepsilon > 0 \quad \exists q = \text{poly}(1/\varepsilon) \text{ s.t.} \]

\[ (\varepsilon, 1) \text{-approx'ing Label-Cover}(q) \text{ is \#P-hard} \]

\[ \Rightarrow \text{ Many inapprox results; e.g. } (\varepsilon/3 + \varepsilon) \text{-approx 3-Lin} \]

\[ (\varepsilon, 1 + \varepsilon) - \text{Max-Independent-Set} \]

\[ \text{\textbf{Raz: 3Sat}} \text{ size } n \rightarrow \text{LC size } \text{\#poly}(\log(n)) \]

\[ \text{ETH} \Rightarrow \text{\#approx 3-Lin reg} \quad (\varepsilon, 1 + \varepsilon) \text{ time}. \]
\[ \text{UGC} : \text{ Ug } = \text{ Label-Cover where there is one bijection} \]

\[ V^* \leq O \text{ s.t. } (5, 1 - \delta) \text{-approx. UG is NP-hard.} \]

Maybe should have said "not in polynomial time."

Many strong inapprox results: whatever

e.g., [Ragh '09] VCSPs \( \text{deg-k Lasserre/Parrilo boost} \)

\[ \text{UGC} = \text{(a, b) - approximates getting } (x^*, \beta - \varepsilon) \text{ is NP-hard.} \]

Basically, assuming UGC closes NP-hardness of all CSPs.

But... [ABS '10] \( V^* \leq O \), \((5, 1 - \delta)\)-approximating UGC doable in \( O(n^{5/3}) \) time.

Assuming ETH, other aspect this is as easy as an NPC problem can be.

UGC de facto easy??

* No known "any-case hard" instances
  * Lasserre/Parrilo \( \text{deg } H \) solves all known "tricky" instances