CS 221 LECTURE 14 3/8/18

TODAY: AVERAGE CASE COMPLEXITY

- Motivation
- Algorithms + Definitions.
- Complexity Results
  - Permanent hard on average (for #P)
  - DNP-Completeness
  - Hardness of "Lattice" Problems.

Motivations:

(1) Empirical Complexity: Naturally occuring instances of problems not always hard. E.g. Many SAT-solvers ... Why? Perhaps problems come from nice distribution?

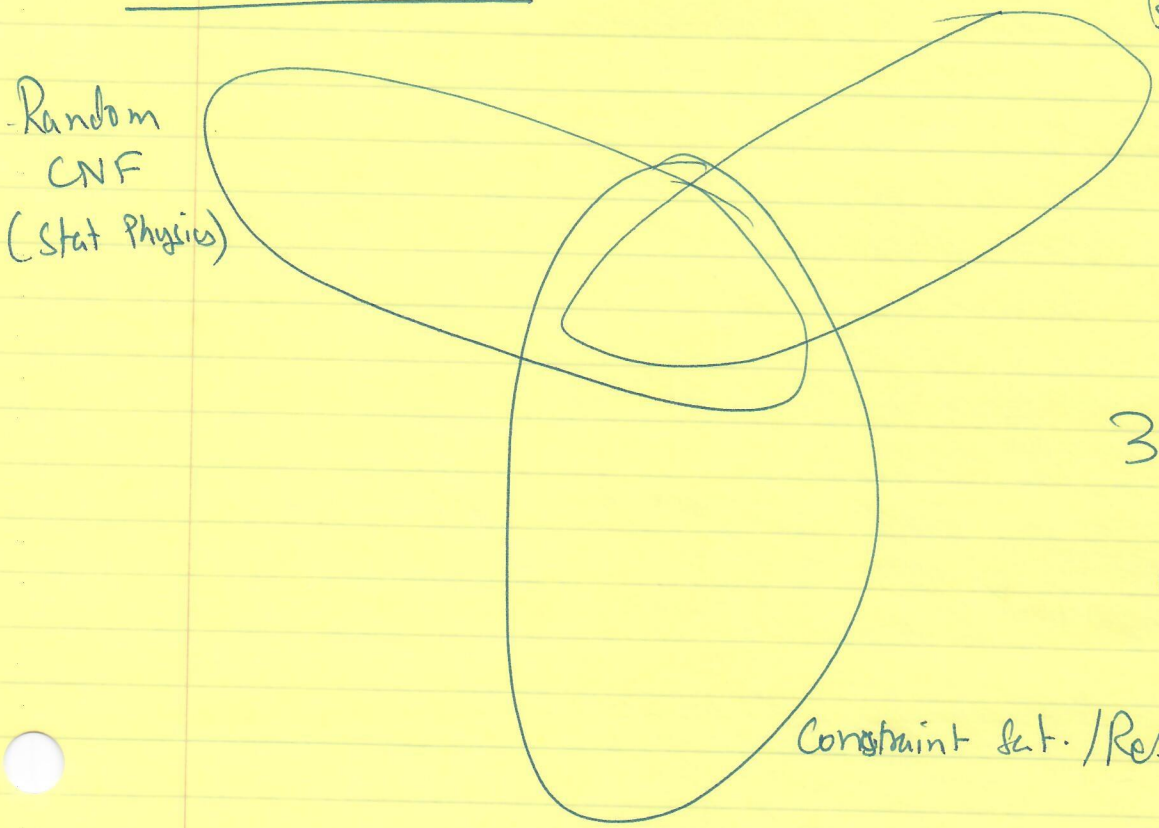
(2) Cryptography: Need to generate hard instances of problems (with known solutions)

Public-key: I know how to decrypt but you don't.

Private-key: Need to know key to decrypt.

[problems ~~either~~ hard to solve even when inputs randomly generated.]

Algorithmic Efforts



3 different communities (at least).

Typical Thms:

① $G_{n,p}$:- W.h.p. $\exists$ clique of size $2\log n$ in $G(n, \frac{1}{2})$
- Whp can find clique $\log n$ in poly time.
- Whp No clique of size $(2+\epsilon)\log n$.

② Random CNF : m Clauses selected independently on n var.

Thm: if $m > \alpha n \Rightarrow$ no sat. assgmt.
$m \leq \beta n \Rightarrow$ whp $\exists$ sat assgmt.
$m \leq \gamma n \Rightarrow$ Can find assgmt
$m \geq f(n) \Rightarrow$ can prove not sat. assgmt

for 4-SAT above: $\alpha = \beta \stackrel{?}{=} \gamma$

What is Easy? [what is Avg-P?]

1st Choice: $\mathop{E}_{x \sim D_n}[T(x)] = \text{poly}(n)$.

Not nice. $T(x) \Rightarrow T^2(x)$

poly $\Rightarrow$ exp.

Better Choices:

2nd Choice [Impagliazzo]: $\exists$ poly $q(\cdot)$ s.t. $\forall$ $p(\cdot)$,

$$\Pr_x[T(x) \geq q(n)] \leq \frac{1}{p(n)}$$

3rd Choice [Levin]: $\exists$ c s.t.

~~$\Pr_x[T(x)]$~~ $\mathop{E}_x[T(x)^{1/c}] = O(n)$.

[Exercise: Definitions Equivalent?]

Distributional Problems

Really a pair (L, D) ; $L \subseteq \{0,1\}^*$

$D = \{D_n\}$ being a collection of distributions one for each n

D_n supported on $\{0,1\}^n$.

Example $(SAT, D_{m,n})$; or $(Clique, G_{n,p})$

or $(Parm, D_{n, p=poly(n)})$ ~~or~~ &

Is SAT hard on average if (1) $\exists$ Distribution on which it is hard?

(2) it is hard on uniform?

My view: Distributions should be "sampleable".

(also large subset of TCS)

D_n is P-Sampleable if $\exists$ poly time deterministic algorithm (circuit) that takes unif. bits as input & produces samples from D_n.

3 Kinds of hardness

(1) "Worst-case hardness rel. to complexity class"

Example Thm: if $(\text{Perm}, U_{n\times n}(\mathbb{Z}_p), p=\text{poly}(n))$ is in average-P, then $\#P \subseteq BPP$.

"Permanent on average is $\#P$-complete"

(2) Worst-case hardness rel. to some problem

Example Thm: [Ajtai], [Regev]: if it is easy to find short vectors in random lattice X, then can find approx. short vectors in any lattice.

[Catch: Nöther problem is "NP-hard"].

(3) Pure avg-case hardness:

if (IL, Unif) in average-P

↑ Impagliazzo-Levin problem

then every (L, D) is in average-P.

↑ in NP ↑ P-samplable

Hardness of Permanent [Lipton].

- Suppose $A(M, p)$ computes $\text{perm}(M) \bmod p$ with prob. $\geq 1 - \frac{1}{n^5}$ when $M \sim \text{Unif.}(\mathbb{Z}_p^{n \times n})$ & $p \sim \text{Unif.}[n^3, 2n^4]$

- Suppose we wish to compute $\text{perm}(N)$ over integers $N \in \{0,1\}^{n \times n}$ N worst-case.

- Idea: Pick $p \sim \text{unif}\ [n^3, n^4]$

$M \sim \text{unif}\ (\mathbb{Z}_p^{n \times n})$

x ← variable.

consider $\text{perm}(N + x \cdot M)$

Claim 1: $f(x) \triangleq \text{perm}(N + xM)$ is a poly in $\mathbb{Z}_p[x]$ of deg $\leq n$.

$$f(x) = \text{perm}\left(\begin{bmatrix} N_{11} + x M_{11} & N_{12} + x M_{12} & \cdots \\ & \alpha + x \cdot \beta & \\ \cdots & & \cdots \end{bmatrix}\right)$$

Claim 2: $f(0) = \text{perm}(M) \bmod p$.

$\forall i$

Claim 3: $\Pr_{\substack{\$ \\ M}}\left[f(i) \neq A(N+i\,M)\right] \leq \frac{1}{n^5}$

Claim 4: $\Pr\left[\exists i \in [1 \ldots n+1] \text{ s.t. } f(i) \neq A(N+i\,M)\right] \leq \frac{1}{n^4}$

But if we know $f(1) \ldots f(n+1) \Rightarrow$ can interpolate $f(0)$.
[Repeat for many primes, apply CRT. $\S$ + 1 more union bound].
if $A \in BPP \Rightarrow$ Above algorithm in BPP!

State-of-the-art:

if $A(M, p) = \text{perm}(M) \bmod p$ w.p. $\geq \frac{1}{\text{poly}(n)}$ then

$\#P \subseteq BPP$.

(Good Project to write clean proof.).

The Regev Problem

- Yes Distribution : Unknown d, [d sampled once & for all]

  Samples = $x \cdot d + e$ [x uniform, e error = gaussian]

- No distribution: Uniform numbers.

- Claim: YES $\approx_p$ NO to polytime algorithms unless SVP in lattices ~~def~~ poly(n)-approximable.

DNP-Completeness:

Reductions $R : \Pi_1 \longrightarrow \Pi_2$ is a reduction from (Π_1, D_1) to (Π_2, D_2)

if $\begin{cases} R(D_1) = D_2 \ ? \\ R(D_1) \approx D_2 \ ? \end{cases}$ } Neither necessary.

$\underbrace{R(D_1)}_{\tilde{D}_1}$ "ϵ-dominates" D_2

$\tilde{D}_1$ ϵ-dominates D_2 if

$$\Pr_{x \sim \tilde{D}_1}\left[D_2(x) < \frac{D_1(x)}{P(|x|)} \right] \leq \epsilon.$$

Thm [LL]: ∃ problem $L \in NP$ s.t.

$(L, Unit)$ is DNP-hard.

(Proof Omitted.)