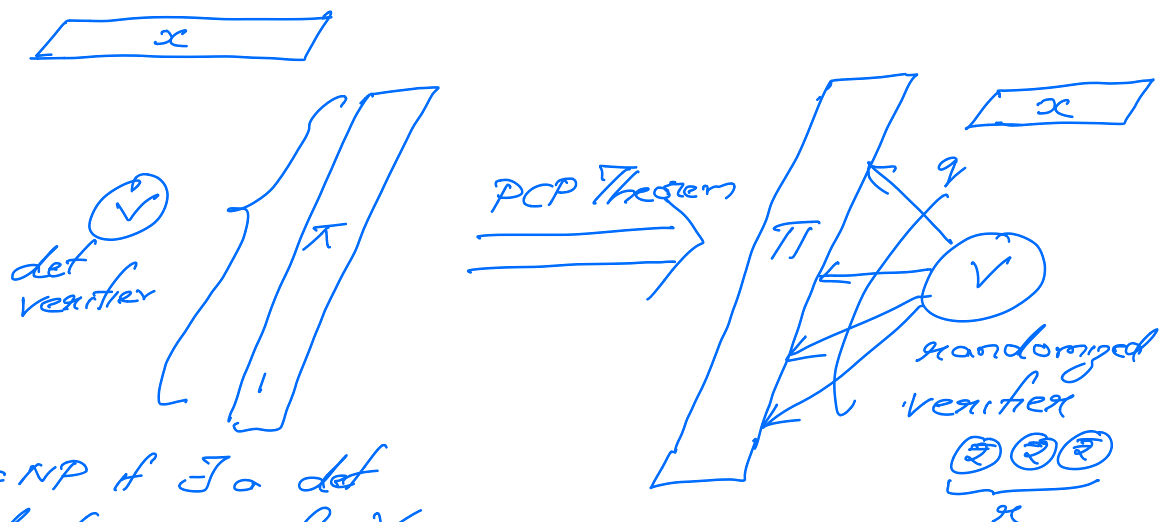
Today
- PCPs
  * Introduction
  * PCPs & Inapproximability

Lecture 25
Computational Complexity (5 May '20)
Instructor: Prahladh Harsha

Recall: (from last time)

MIP = NEXP
$\downarrow$ Scaling down
PCP Theorem.

Recall classical defn of NP
$L \in NP$



$L \in NP$ if $\exists$ a det poly time verifier V s.t

$x \in L \Rightarrow \exists \pi,\ V(x,\pi) = 1$

$x \notin L \Rightarrow \forall \pi,\ V(x,\pi) = 0$

Key differences:

(1) Strengthen verifier
det $\rightarrow$ randomized

(2) Weaken verifier
Reading proof entirely
$\rightarrow$ proof in $\leq q$ locations

(1)

Formal Defn of verifier.

Definition: (r, q, m, t, a)-restricted verifier

$r, q, m, t, a : \mathbb{N} \to \mathbb{N}$

is a prob TM that on input x of length n and oracle access to a proof π

- tosses at most $r(n)$ random coins R
- queries the proof π in at most $q(n)$ locations ($|\pi| = m(n)$).
- runs in time $t(n)$.
- computes a predicate D of size $a(n)$
- Accepts if the proof bits it reads satisfies the predicate D.

$$V(x, R) \to (\underbrace{Q}_{\text{of locations}}, D)$$

PCP class:

$r, q, m, t, a : \mathbb{N} \to \mathbb{N}$. $\quad 0 \le s < c \le 1$

$L \in PCP_{c,s}[r, q, m, t, a]$ if $\exists$ a (r, q, m, t, a)-restricted verifier V s.t

Comp: $x \in L \Rightarrow \exists \pi \quad \Pr_R[D(\pi|_Q) = 1] \ge c$

Sound: $x \notin L \Rightarrow \forall \pi, \quad \Pr_R[D(\pi|_Q) = 1] \le s$.

$$NP = \bigcup_a PCP_{1,0}[0, n^a, n^a, n^a, n^a]$$

We will usually drop parameter m, t, a

r – randomness

q – query complexity

$$NP = \bigcup_a PCP_{1,0}[0, n^a]$$

$$BPP = \bigcup_a PCP_{2/3, 1/3}[n^a, 0]$$

$m \le q(n) \cdot 2^{r(n)}$, $t(n) = poly(n)$, $a(n) = poly(q)$

$c = 1$; perfect completeness.

PCP Theorem I: There exists a constant Q, $\forall L \in NP$, $\exists$ constant c.

$$L \in PCP_{1, 1/2}[\underbrace{c \log n}_{\text{toss random coins}}, \underbrace{Q}_{\text{Q locations of the proof.}}]$$

$PCP_{c,s}[r, q] \subseteq NTIME(\cdot 2^r(t+a))$

(by going over all randomness)

PCP Theorem: $NP \subseteq \bigcup_c PCP_{1, 1/2}[c \log n, Q] \subseteq NP$

In short $NP = PCP_{1,1/2}[\log, O(1)]$

Approximation Algorithms

NP-hard combinatorial optimization problem

- Vertex Cover
- Max Sat
- Max Clique
- TSP

} NP-hard to compute the optimal soln.

- What about an approximation.

$\alpha \in (0,1)$. Φ - Maximization Problem

A is an α-approximation for the Φ if for every instance x of Φ

$$x \longrightarrow \boxed{A} \longrightarrow A(x)$$

$$\alpha \cdot OPT_{\Phi}(x) \le A(x) \le OPT_{\Phi}(x)$$

<u>Vertex Cover:</u> $\frac{1}{2}$-approximation

Find a maximal matching in G & o/p all end points.

$$|VC(G)| \le 2|MM(G)| \le 2|VC(G)|$$

MAX3SAT: 7/8 - approximation.

$$\psi = \underbrace{(x_i \vee x_j \vee \overline{x_k})(\ldots)}_{m}$$

Find an assign that satisfies as many clauses as possible?

TSP: 3/2 - approximation.

⋮

Qn: Given an max/min problem, what is the best factor upto which we can approximate it?

Focus attention of MAX3SAT

α-approximate of MAX3SAT

↪ Decision Version of the problem.

Defn: gap$_\alpha$-MAX3SAT $(\alpha \in (0,1))$

YES $= \{(\varphi, k) \mid \exists$ an assignment that satisfies at least k clauses$\}$

NO $= \{(\varphi, k) \mid \forall$ assign at most αk clauses are satisfied$\}$.

Prop: α-approximation of MAX3SAT is polytime
$\Updownarrow$
gap_α-MAX3SAT is polytime.

Pf: ($\Rightarrow$) Suppose A is an α-approx alg for MAX3SAT

B = On input (φ, k)
1. Run A on φ & let $k' = A(\varphi)$
2. Accept if $k' \geq \alpha k$.

$(\varphi, k) \in YES \Rightarrow OPT(\varphi) \geq k \Rightarrow A(\varphi) \geq \alpha \cdot OPT(\varphi)$
$\Rightarrow A(\varphi) \geq \alpha k$
$\Rightarrow B$ is correct

$(\varphi, k) \in NO \Rightarrow OPT(\varphi) \leq \alpha k \Rightarrow A(\varphi) \leq OPT(\varphi) \leq \alpha k$
$\Rightarrow B$ is correct.

($\Leftarrow$) Suppose B is an alg for gap_α-MAX3SAT

A: On input φ.
1. m = #clauses of φ
2. Run $B(\varphi, 1), \ldots, B(\varphi, m)$.
3. k - largest s.t. $B(\varphi, k)$ = yes
4. Output αk.

B rejects $(\varphi, k+1)$ & B accepts (φ, k)

$OPT(\varphi) \leq k \qquad\qquad OPT(\varphi) \geq \alpha k$

Hence, $\alpha \cdot OPT(\varphi) \leq \alpha k \leq OPT(\varphi)$

Hence, A is correct $\boxtimes$

PCP Theorem II:

$\exists$ an $\alpha \in (0,1)$ such that SAT is polytime is reducible to gap_α-MAX3SAT

R

$$\psi \in SAT \Rightarrow R(\psi) = (\varphi, k) \in YES$$
$$\psi \notin SAT \Rightarrow R(\psi) = (\varphi, k) \in NO$$

Cor: $\exists\ \alpha \in (0,1)$, α-approximating MAX3SAT is NP-hard.

gap_α-MAXSAT*

YES = $\{\varphi \mid \varphi \in 3SAT\}$

NO = $\{\varphi \mid \forall$ assignments at most $\alpha \cdot m$ clauses are satisfied$\}$

PCP Theorem III: $\exists\ \alpha \in (0,1)$

gap_α-MAX3SAT* is NP-hard. (under deterministic redns)

Obs: PCP Thm III $\Rightarrow$ PCP Thm II

Lemma: PCP Theorem I & PCP Theorem III are equivalent.

Equivalence of the 2 versions of PCP Theorem

Pf: ($\Leftarrow$) PCP Thm III $\Rightarrow$ PCP Thm I.

Suppose gap_{α}-MAX3SAT* is NP-hard.

& let $L \in NP$

$L \leq_p gap_{\alpha}$-MAX 3SAT*.

Restricted Verifier: V: On input x

π – assignment to the vars of $R(x) = \varphi$.

1. It runs redn R to compute $\varphi = R(x)$
2. Pick a random clause C in φ.
3. Query the 3 vars in C & accept if the proof satisfies the clause.

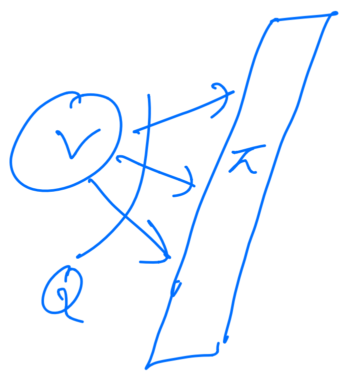
$L \in PCP_{1,\alpha}[\log m, 3]$, m = #clauses in $R(x) = \varphi$

$\subseteq PCP_{1,\alpha^k}[k \log m, 3k]$

($\Leftarrow$) PCP Theorem I $\Rightarrow$ PCP Theorem III.
SAT

(PTI) $\Rightarrow$ SAT has a $(c \log n, Q)$ restricted verifier for some constant c & Q.

We need to find a red$\underline{n}$ from SAT to gap$_{1/\alpha}$-MAX3SAT*



MAXQSAT
- Vars: proof bits of the restricted verifier
- Clauses:
$R \longrightarrow Q$ locations
D_R

$$\Phi = \bigwedge_R \underbrace{D_R}_{\leq Q \text{ arity.}}$$ — MAXQSAT instance

Running time of red$\underline{n}$ $= 2^{r(n)} \cdot t(n)$.
$= 2^{c \log n} \cdot \text{poly}(n)$
$= \text{poly}(n)$.

MAXQSAT $\longrightarrow$ MAX3SAT.

Obs: $\forall$ Q, $\exists$ $l(Q)$ & $k(Q)$ s.t any Boolean fn φ on Q variables can be encoded by a 3CNF formula $h(\varphi)$ w/ $l(Q)$ - extension variables & $k(Q)$ clauses

s.t

φ is sat $\Rightarrow$ $h(\varphi)$ is satisfiable

φ is not sat $\Rightarrow$ $h(\varphi)$ is not satisfiable

—

$$\Phi = \bigwedge_{k} h(D_k)$$

MAX3CNF instance.

YES: Φ is a satisfiable 3SAT instance

NO: $\forall$ assign. # clauses are satisfied

$$= (1-\alpha)\cdot 1 + \alpha\left(1-\frac{1}{k}\right)$$

$$= 1-\frac{\alpha}{k}.$$

Polynomial time redn

SAT to $gap_{1,\alpha/k}$ - MAX3SAT

$\boxtimes$

gap$_\alpha$-MAX3AT* is NP-hard

$\Updownarrow$

restricted verifier for NP

→ PCP Thm.

$$\boxed{\text{SAT} \le_p \text{gap}_\alpha\text{-MAX3SAT}}$$

<u>Cor</u> ① If there is a det α-approx for MAX3SAT $\Rightarrow$ SAT $\in$ P

② If there is a rand α-approx for MAX3SAT $\Rightarrow$ SAT $\in$ BPP

(Next time: Inapproximability of Clique, Sketch of PCP Thm Proof).